

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITECNICA SUPERIOR



INGENIERIA TECNICA EN INFORMATICA DE GESTION

PROYECTO FIN DE CARRERA
DESARROLLO DE UN PORTAL WEB PARA LA
RECUPERACION DE LIBRERIAS DLL

Tutor: Jorge Morato Lara

Autor: José Luis Barral Paniagua

A todas las personas que me han apoyado durante todos estos años, y me han ayudado a creer en mí para conseguir este objetivo que parecía inalcanzable.

A mis padres, José Luis y Amparo, a mis hermanas María de los Ángeles y Laura, a mi tutor Jorge, a Rocío que me apoyó en todo momento y a Mariasun, quien me convenció de que yo podía con esto y con cualquier otra cosa que me propusiera.

Gracias a todos.

Resumen

Una biblioteca o librería es un compendio de subprogramas utilizado para facilitar el desarrollo del software. Las bibliotecas contienen código y datos que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular.

Una biblioteca o librería de enlace dinámico (DLL) es un archivo propio de Windows que podrá ser llamado desde cualquier aplicación Windows. De hecho, se puede considerar que Windows está construido sobre una gran cantidad de DLL's.

La utilidad de un repositorio de DLL's radica en que en programación la optimización del tiempo es fundamental, y el primer paso antes de desarrollar cualquier aplicación es investigar si existe un precedente, es decir si ya hay algo hecho en el mismo campo, y si es así aprovecharlo.

La segunda utilidad surge porque es relativamente frecuente en Windows que surjan problemas con archivos DLL por razones que veremos, y debido a ello nos veamos en la necesidad de tener que conseguir cierta DLL concreta en un determinado momento porque ha dejado de funcionar una aplicación, o el sistema operativo en el peor de los casos.

Este trabajo surge porque aunque actualmente existen algunos repositorios, ninguno facilita una búsqueda completa por todas las propiedades de una DLL, con lo cual actualmente estos archivos no son fácilmente localizables.

El objetivo por tanto es la realización de una herramienta que supere a las que actualmente existen en este campo recogiendo lo mejor de cada una, además de ofrecer en lo posible alguna funcionalidad adicional que ninguna ofrezca.

Pensamos que el trabajo presentado en efecto cumple con lo expuesto en esta introducción, es decir, por un lado supera funcionalmente a todos los repositorios investigados en el estado del arte, como veremos, y en segundo lugar consideramos que es una herramienta útil tanto para programadores como para usuarios de Windows.

Tabla de Contenido

Resumen	3
Índice de ilustraciones	7
Índice de tablas	9
1. Introducción.....	11
1.1 Motivación	11
1.2 Objetivos	15
1.3 Resto del documento	15
2. Estado del Arte	16
2.1 Repositorio de Publispain	16
2.2 Repositorio “My-DLL”	18
2.3 Repositorio “Archivos-DLL”	20
2.4 Repositorio “DLL-Download-System”	22
2.5 Repositorio “DLL-Files”	26
2.6 Repositorio “A Free DLL”	28
2.7 Repositorio “DLL-Downloads”	30
2.8 Comparación de los repositorios	34
3. Marco del proyecto	37
3.1 Paradigma de desarrollo utilizado	37
3.2 Metodología de desarrollo utilizada	37
3.2.1 Adaptación del Proceso Unificado a las necesidades del producto	38
3.2.2 El Proceso Unificado y la metodología SunTone	43
4. Fase de Comienzo	44
4.1 Vista general del proyecto	44
4.1.1 Introducción	44
4.1.2 Solución propuesta	44
4.1.3 Riesgos	46
4.1.4 Restricciones	46
5. Fase de Elaboración	47
5.1 Flujo de Requisitos	47
5.1.1 Introducción	47
5.1.2 Restricciones y Suposiciones	49
5.1.3 Riesgos	50
5.1.4 Restricciones	50
5.1.5 Requisitos funcionales	51
5.1.6 Requisitos no funcionales	71

5.1.7	Glosario del Proyecto	76
5.2	Flujo de Análisis	81
5.2.1	Análisis de los casos de uso	81
5.2.2	Modelo de negocio	97
5.3	Flujo de Diseño	108
5.3.1	Modelo de diseño para C1-BuscarArchivos	109
5.3.2	Modelo de diseño para C2-VerArchivos	113
5.3.3	Modelo de diseño para C3-DescargarArchivo	115
5.3.4	Modelo de diseño para C4-Registrarse	117
5.3.5	Modelo de diseño para C5-ModificarRegistro	119
5.3.6	Modelo de diseño para C6-Autenticarse.....	121
5.3.7	Modelo de diseño para C7-SubirArchivo	123
5.3.8	Modelo de diseño para C8-EliminarArchivo.....	125
5.3.9	Modelo de diseño para C9-ModificarArchivo.....	127
5.3.10	Modelo de diseño para C10-SubirComentarioArchivo	129
5.4	Arquitectura del software. Modelo de arquitectura	131
5.4.1	Diagrama de despliegue de alto nivel	132
5.4.2	Diagrama de despliegue detallado	134
5.4.3	Diagrama de despliegue de niveles y capas.....	137
5.5	Flujo de implementación.....	140
5.5.1	Indexación por medio de Lucene.....	140
5.5.2	Invocación al API de Windows 32-bit por medio de Jawin	142
5.6	Flujo de implantación.....	145
5.6.1	Requisitos	145
5.6.2	Ficheros de instalación	145
5.6.3	Procedimiento	146
5.6.4	Archivos de configuración de la aplicación.....	148
6.	Fase de Transición. Evaluación	150
6.1	Herramientas y condiciones	150
6.2	Escenario	151
6.3	Descripción de las pruebas.....	154
6.4	Resultados de las pruebas	155
6.4.1	Prueba 1: 1 grupo de hilos con 1 usuario	155
6.4.2	Prueba 2: 2 grupos de hilos con 1 usuario por hilo.....	157
6.4.3	Prueba 3: 4 grupos de hilos con 1 usuario por hilo.....	158
6.4.4	Prueba 4: 8 grupos de hilos con 1 usuario por hilo.....	160
6.4.5	Prueba 5: 8 grupos de hilos con 2 usuarios por hilo	162
6.4.6	Prueba 6: 8 grupos de hilos con 3 usuarios por hilo	164
6.5	Resumen y observaciones	166
6.5.1	<i>Tiempo de respuesta</i>	166
6.5.2	<i>Rendimiento</i>	168
7.	Planificación y coste del proyecto	170
7.1	Planificación.....	170

7.2	Coste del proyecto.....	171
7.2.1	Coste de personal.....	171
7.2.2	Coste de los equipos	173
7.2.3	Costes indirectos	173
7.2.4	Riesgo	174
7.2.5	Total Coste	174
8.	Conclusiones y Trabajos Futuros.....	175
8.1	Conclusiones	175
8.1.1	Producto	175
8.1.2	Proceso.....	175
8.1.3	Personales	176
8.2	Trabajos futuros	177
8.2.1	Mejoras del prototipo/sistema presentado	177
8.2.2	Otros ámbitos en los que aplicar el trabajo realizado	178
9.	Bibliografía	179
10.	Apéndices	180
10.1	Apéndice 1: Solucionar problemas típicos con DLL.....	180

Índice de ilustraciones

Ilustración 1- Repositorio Publispain, http://www.publispain.com/dlls	16
Ilustración 2- Repositorio MY-DLL, http://www.my-dll.com/index.php?lang=sp	18
Ilustración 3- Repositorio Archivos DLL, http://www.archivos-dll.es/	20
Ilustración 4- Repositorio DLL-Download-System	22
Ilustración 5- Búsqueda de Dll-download-system	23
Ilustración 6- Pagina de resultados de DLL-download-system	24
Ilustración 7- RepositorioDLL-Files, http://www.dll-files.com/	26
Ilustración 8- Pagina de resultados de DLL-files.....	27
Ilustración 9- Repositorio AfreeDLL, http://www.afreedll.com/	28
Ilustración 10- Información de archivo de AfreeDLL.....	29
Ilustración 11- Repositorio Dll-downloads, http://www.dll-downloads.com	30
Ilustración 12- Página para descargar Archivo de DLL-downloads	31
Ilustración 13- Pagina para elegir sistema operativo en DLL-downloads	32
Ilustración 14- Pagina de búsqueda en DLL-downloads	33
Ilustración 15- Diagrama de casos de uso del sistema.....	53
Ilustración 16- Diagrama de clases de análisis	97
Ilustración 17- Tipos de componentes de diseño	108
Ilustración 18- Diagrama secuencia de la búsqueda facetada	110
Ilustración 19- Diagrama de secuencia para la Búsqueda Por Sistema Operativo.....	111
Ilustración 20- Diagrama de secuencia para mostrar página siguiente	112
Ilustración 21- Diagrama de secuencia para el caso de uso C2-VerArchivos	114
Ilustración 22- Diagrama de secuencia para el caso de uso C3-DescargarArchivo	116
Ilustración 23- Diagrama de secuencia para el caso de uso C4-Registrarse	118
Ilustración 24- Diagrama de secuencia para el caso de uso C5-ModificarRegistro.....	120
Ilustración 25- Diagrama de secuencia para el caso de uso C6-Registrarse	122
Ilustración 26- Diagrama de secuencia para el caso de uso C7-SubirArchivo.....	124
Ilustración 27- Diagrama de secuencia para el caso de uso C8-EliminarArchivo	126
Ilustración 28- Diagrama de secuencia para el caso de uso C9-ModificarArchivo	128
Ilustración 29- Diagrama de secuencia para C10-SubirComentarioArchivo	130
Ilustración 30- Esquema del patrón Modelo-vista-controlador	131
Ilustración 31- Diagrama de la arquitectura general del sistema	133
Ilustración 32- Diagrama de despliegue detallado del sistema	136

Ilustración 33- Diagrama niveles y capas del sistema	139
Ilustración 34- Gráfica del tiempo de respuesta de la prueba 1	156
Ilustración 35- Gráfica del tiempo de respuesta de la prueba 2	157
Ilustración 36- Gráfica del tiempo de respuesta de la prueba 3	159
Ilustración 37- Gráfica del tiempo de respuesta de la prueba 4	161
Ilustración 38- Gráfica del tiempo de respuesta de la prueba 5	163
Ilustración 39- Gráfica del tiempo de respuesta de la prueba 6	165
Ilustración 40- Gráfica de la media del tiempo de respuesta	167
Ilustración 41- Gráfica de la mediana del tiempo de respuesta	168
Ilustración 42- Gráfica del rendimiento en muestras por segundo	169
Ilustración 43- Diagrama de Gantt del sistema	171

Índice de tablas

Tabla 1- Comparativa de los repositorios de archivos dll.....	35
Tabla 2- Adaptación del Proceso Unificado a nuestro producto.....	42
Tabla 3- Casos de uso del sistema.....	54
Tabla 4- Requisitos funcionales del sistema.....	58
Tabla 5- Análisis caso de uso C1-BuscarArchivos.....	81
Tabla 6- Análisis del caso de uso C2-VerArchivos.....	83
Tabla 7- Análisis del caso de uso C3-DescargarArchivo.....	84
Tabla 8- Análisis del caso de uso C4-Registrarse.....	85
Tabla 9- Análisis del caso de uso C5-ModificarRegistro.....	87
Tabla 10- Análisis del caso de uso C6-Autenticarse.....	89
Tabla 11- Análisis del caso de uso C7-SubirArchivo.....	91
Tabla 12- Análisis del caso de uso C8-EliminarArchivo.....	93
Tabla 13- Análisis del caso de uso C9-ModificarArchivo.....	94
Tabla 14- Análisis del caso de uso C10-SubirComentarioArchivo.....	95
Tabla 15- Definición de la clase SrvArchivo.....	98
Tabla 16- Definición de la clase SrvRepositorio	99
Tabla 17- Definición de la clase SrvBuscador.....	99
Tabla 18- Definición de la clase SrvIndizador.....	100
Tabla 19- Contenido de la clase SrvUsuario.....	100
Tabla 20- Contenido de la clase SrvComentario.....	101
Tabla 21- Contenido de la clase Archivo.....	101
Tabla 22- Contenido de la clase TipoArchivo.....	103
Tabla 23- Contenido de la clase SistemaOperativo.....	103
Tabla 24- Contenido de la clase BusquedaArchivos.....	104
Tabla 25- Contenido de la clase PaginaCodigos.....	105
Tabla 26- Contenido de la clase Lenguaje.....	106
Tabla 27- Contenido de la clase Comentario.....	106
Tabla 28- Contenido de la clase Usuario.....	107
Tabla 29- Resultados de la prueba de rendimiento 1.....	156
Tabla 30- Resultados de la prueba de rendimiento 2.....	157
Tabla 31- Resultados de la prueba de rendimiento 3.....	158
Tabla 32- Resultados de la prueba de rendimiento 4.....	160

Tabla 33- Resultados de la prueba de rendimiento 5.....	162
Tabla 34- Resultados de la prueba de rendimiento 6.....	164
Tabla 35- Resumen de los resultados de las pruebas de rendimiento.....	166
Tabla 36- Distribución del tiempo por fases de desarrollo.....	170
Tabla 37- Distribución de tareas por trabajador.....	172
Tabla 39- Calculo del coste de personal.....	172
Tabla 40- Coste de los equipos hardware.....	173
Tabla 41- Coste total del proyecto.....	174

1. Introducción

En la introducción presentaremos el contexto y la motivación del proyecto, así como los objetivos principales que nuestro sistema quiere cumplir. Por último se da un resumen de los aspectos tratados en cada capítulo restante de este documento.

1.1 Motivación

DLL es el acrónimo de Dynamic Link Library (Librería de enlace dinámico), es decir una librería de funciones ejecutables o de datos. Es exclusiva del sistema operativo Windows. Puede ser usada por aplicaciones de dicho sistema. Una DLL proporciona una o más funciones determinadas. Los programas acceden a las funciones creando un enlace dinámico o uno estático a la DLL. Un enlace estático es permanente mientras dura la ejecución del programa mientras que un enlace dinámico lo crea el programa según lo necesita y luego lo destruye. Las DLL pueden contener solo datos. Generalmente estas bibliotecas llevan la extensión “.dll”.

Las DLL reciben a veces el nombre de ejecutables sin tarea debido a que son dirigidos por la aplicación (o tarea). Todo lo que precisa una DLL lo pide a la aplicación. Es por esto que las DLL no tienen significado ni función por sí mismas.

El propósito principal de una DLL es actuar como librería permanentemente disponible de funciones o recursos.

Varias aplicaciones pueden usar una DLL al mismo tiempo. El Sistema Windows proporciona algunas DLL que están disponibles para cualquier aplicación basada en Windows. Otras DLL son creadas para una aplicación específica y cargadas cuando la aplicación se está ejecutando.

Las aplicaciones difieren de las DLL en que pueden funcionar simultáneamente múltiples copias o instancias de una aplicación [1].

En una instalación típica de Windows las DLL se encuentran normalmente en los siguientes directorios según el sistema operativo:

- **XP** - C:\Windows\System32.
- **95/98/Me** - C:\Windows\System.

- **NT / 2000** - C:\WINNT\System32 [2].

Un archivo DLL presenta tres propiedades principales:

- **Versión:** Muy importante para evitar problemas que más adelante veremos.
- **Descripción:** que nos indica para que vale el archivo.
- **Copyright:** que nos indica que empresa lo ha desarrollado.

Su uso proporciona algunas ventajas:

- Facilitan la reutilización de código, eliminando la implementación reiterada de una tarea frecuente.
- Reducen el tamaño de los archivos ejecutables: Gran parte del código puede estar almacenado en bibliotecas y no en el propio ejecutable lo que redundaría en una mejor modularización.
- Pueden estar compartidas entre varias aplicaciones: Si el código es suficientemente genérico, puede resultar de utilidad para múltiples aplicaciones (por ejemplo, la MFC es una biblioteca dinámica con clases genéricas que recubren la API gráfica de Windows y que usan gran parte de las aplicaciones).
- Facilitan la gestión y aprovechamiento de la memoria del sistema: La carga dinámica permite al sistema operativo aplicar algoritmos que mejoren el rendimiento del sistema cuando se carguen estas bibliotecas. Además, al estar compartidas, basta con mantener una copia en memoria para todos los programas que la utilicen.
- Brindan mayor flexibilidad frente a cambios: Es posible mejorar el rendimiento o solucionar pequeños errores distribuyendo únicamente una nueva versión de la biblioteca dinámica. Nuevamente, esta corrección o mejora será aprovechada por todas las aplicaciones que compartan la biblioteca [3].

Pero también presentan inconvenientes. En Windows estas bibliotecas están muy extendidas y son compartidas por múltiples aplicaciones. De esta compartición de código surgen dos problemas que constituyen lo que se ha denominado “infierno de las DLL”:

- Al instalar un programa se puede reemplazar una DLL por otra versión incompatible (conflicto de versiones).
- Al desinstalar un programa se puede borrar una DLL compartida.

En ambos casos los programas que compartan la DLL dejarán de funcionar.

Para evitar estos problemas se incorpora el número de versión a las DLL para evitar sobrescribirlas con versiones incompatibles, es decir que habrá varias versiones de la misma DLL en el disco duro.

Las versiones más modernas de Windows proponen soluciones automáticas para el problema (mayor control del versionado, petición al usuario de qué DLL quiere mantener...) pero las antiguas instalaciones siguen corrompiendo las DLL. Por ello se han planteado dos soluciones:

- **Scripts de instalación MSI:** se trata de pequeñas bases de datos que indican qué ficheros y versiones instala una aplicación. De esta forma es posible determinar qué versiones son compatibles y cuales no o volver a instalar versiones en el caso de instalaciones corruptas.
- **Ensamblados:** Es un concepto que aparece con .NET y consiste en un fragmento de código ejecutable (DLL o EXE) junto a un fichero (que puede estar incorporado como recurso en el fichero) que indica que contiene, versiones... De esta forma es posible instalar varias versiones diferentes del mismo ensamblado y cargar la que necesita cada programa usando esa información [4].

Estos problemas redundan con relativa frecuencia en que nos aparezca un mensaje de error, en Windows XP, Windows 98, Windows 2000, Windows NT o Windows Vista, sobre un archivo DLL que falta, que no se encuentra o que está dañado, lo cual nos impide ejecutar una aplicación (ver apéndice 1).

Por otra parte, las DLL deberán ser usadas por el programador para almacenar todo el código común a varias aplicaciones. Numerosos expertos sostienen que el código reusado puede suponer hasta el 80 por ciento del total en determinadas circunstancias [1]. No es difícil imaginar lo ventajoso que sería para muchos programadores disponer de ciertas DLL desarrolladas anteriormente que ya implementen las funcionalidades que necesitan para su programa.

Hay tres tipos especiales de DLL. Los archivos OCX, DRV y FON:

OCX es el acrónimo de OLE Custom Control (Control OLE a medida). OLE es el acrónimo de Object Linking and Embedding (enlace e incrustación de objetos). Es un módulo de programa independiente al que pueden acceder otros programas en un entorno Windows. Dicho módulo también es llamado “Control Activex”. Los controles ActiveX u OCX tienen la extensión .ocx. Estos controles representan la segunda generación de Microsoft de control de arquitectura. La primera generación consistía en los controles VBX escritos en Visual Basic.

Los archivos FON, con extensión .fon, son archivos de tipos de letra de Windows. Son también archivos DLL.

Un driver (DRV) es un programa que interactúa con un dispositivo particular o con un tipo especial de software, un software que a menudo es opcional. El driver contiene un conocimiento especial del dispositivo o del software especial, mientras que los programas que usan el driver no tienen dicho conocimiento. Los drivers a menudo están empaquetados como una DLL con extensión .dll en vez de con extensión .drv [6].

Los drivers además presentan otros problemas que surgen cuando queremos instalar un dispositivo antiguo del que no encontramos el driver correspondiente, o queremos instalar un sistema operativo nuevo a un equipo un poco antiguo y no encontramos alguno de los drivers.

El proyecto surge por la utilidad que tendría tener un repositorio que pudiera hacer una búsqueda facetada de archivos DLL. Este repositorio solucionaría muchos de los problemas mencionados, al facilitar la localización y caracterización de estos recursos.

1.2 Objetivos

El objetivo que se persigue es crear un site que almacene archivos DLL, permitiendo su recuperación facetada. Además permitirá incorporar nuevos recursos manualmente. También se pretende superar en funcionalidades a los recursos de naturaleza similar que existen en el mercado.

Para ello primero vamos a estudiar diferentes repositorios de DLL existentes en la Web, analizando sus carencias y posibles mejoras en funcionalidad y usabilidad.

1.3 Resto del documento

En el resto de los capítulos se tratarán los siguientes temas:

- Cap. 2. Estado del Arte: donde se plasma el estudio realizado de las aplicaciones existentes para fijar los objetivos del sistema desarrollado, a partir de las carencias de dichas aplicaciones.
- Cap. 3. Marco del proyecto: donde se da una introducción a la programación orientada a objetos y al Proceso Unificado de desarrollo, que son, respectivamente, la filosofía y la metodología de desarrollo que enmarcan este trabajo.
- Cap. 4. Fase de comienzo: donde se trata la vista general del proyecto, es decir qué solución se propone a partir del estado del arte.
- Cap. 5. Fase de elaboración: donde se expone al análisis, diseño, arquitectura, implementación e implantación del sistema desarrollado.
- Cap. 6. Evaluación: donde se refleja el resultado de las pruebas de carga hechas para verificar los requisitos no funcionales de la aplicación.
- Cap. 7 Planificación y coste: donde se expone la estimación, planificación y costes del proyecto.
- Cap. 8. Conclusiones y trabajos futuros: donde se habla de las conclusiones del producto, del proceso y personales, así como de las mejoras que se podrían realizar de este producto y de la orientación de la posible continuación del mismo.

2. Estado del Arte

A continuación se hace una enumeración y un análisis de los repositorios de DLL disponibles en la Web. Como crítica general hay que destacar que todos ellos tienen el objetivo de proponer un gran número de archivos DLL para ser descargados por cualquiera que los necesite para determinada aplicación o cuando se haya deteriorado el software del ordenador.

2.1 Repositorio de Publispain



Ilustración 1- Repositorio Publispain, <http://www.publispain.com/dlls>

El repositorio de **Publispain** está incluido en la Web de ocio <http://www.publispain.com>.

Este recurso muestra un alfabeto. Cada letra es la inicial de una lista de DLL. Es decir, no hay búsqueda, solo se puede acceder a cada listado mediante el hipervínculo de la inicial del nombre del archivo.

Los archivos se descargan en formato Zip, aunque solo incluye la DLL. El usuario no puede subir archivos. Al descargar un archivo no se presenta ninguna información del mismo. Habría que descargarlo para ver sus propiedades.

2.2 Repositorio “My-DLL”



Ilustración 2- Repositorio MY-DLL, <http://www.my-dll.com/index.php?lang=s p>

El repositorio **my-dll** tiene las siguientes secciones:

- “Home”: presenta un recuadro de búsqueda. Solo se puede buscar por el nombre del archivo. Es necesario poner el nombre completo del archivo, extensión incluida. Cualquier omisión en el nombrado del fichero impide la localización del mismo.
- “Lista de DLLs”: presenta la misma búsqueda que en “Home” y además la lista completa de DLL en una sola página; por tanto resulta poco manejable. Tampoco se puede buscar por inicial.
- “Socios”: Simplemente es una lista de enlaces a sitios Web con quienes tienen buenas relaciones.
- “Contacto”: Es un formulario en el que se pueden realizar consultas, advertir algunos problemas o proponer un archivo DLL para su inclusión. En esta última opción sólo podemos proponerla, pero no subir nuestro archivo.
- “A propósito”: Sección que describe el sitio Web.

Los archivos se descargan en formato Zip e incluyen la DLL y un documento de ayuda.

Al descargar los archivos no te muestra ninguna información de los mismos. Tendremos que descargarlo para ver sus propiedades. Un gran inconveniente es que hay que conocer previamente el nombre del fichero para poderlo descargar. En ningún caso se ordenan los ficheros por funcionalidad ni entorno para el que están desarrollados.

2.3 Repositorio “Archivos-DLL”



Ilustración 3- Repositorio Archivos DLL, <http://www.archivos-dll.es/>

En la página principal presenta una explicación de lo que es una DLL de Windows y de los problemas que hay cuando se deterioran estas DLL y sustituirlas descargándolas de la página. La página presenta también un cuadro de búsqueda, la lista de las DLL más buscadas y una lista de las nuevas recientemente incluidas.

En el recuadro de búsqueda solo se puede buscar por el nombre del archivo, pero no hace falta ponerlo entero. Basta con poner una parte.

Al descargar un archivo presenta la siguiente información del mismo: nombre, descripción, versión y tamaño. No presenta el copyright del archivo. Tendremos que descargarlo para verlo. El tipo de archivo descargado es directamente un .DLL no un Zip. Presenta también una lista de comentarios de usuarios a cada archivo.

El sitio contiene además las siguientes secciones:

- “Todas las DLL”: Te presenta todas las DLL del repositorio en varias páginas. Se puede acceder a cada una por la inicial del nombre, presentándose en pantalla el listado de las DLL que comienzan por esa letra.
- “Contacto”: formulario para mandar cualquier comentario, no se pueden subir archivos.

Como en otros casos la búsqueda presupone un conocimiento previo de la DLL.

2.4 Repositorio “DLL-Download-System”

Su dirección web es <http://es.dll-download-system.com/>

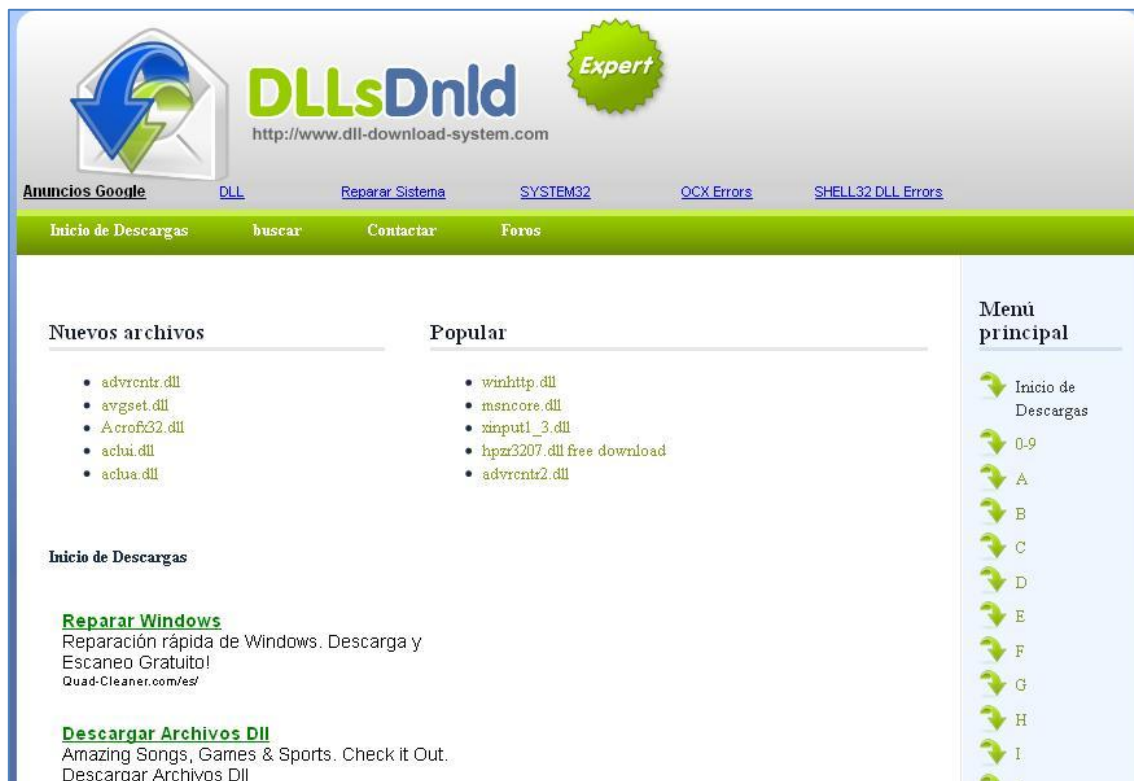


Ilustración 4- Repositorio DLL-Download-System

Las secciones más importantes que presenta son las siguientes:

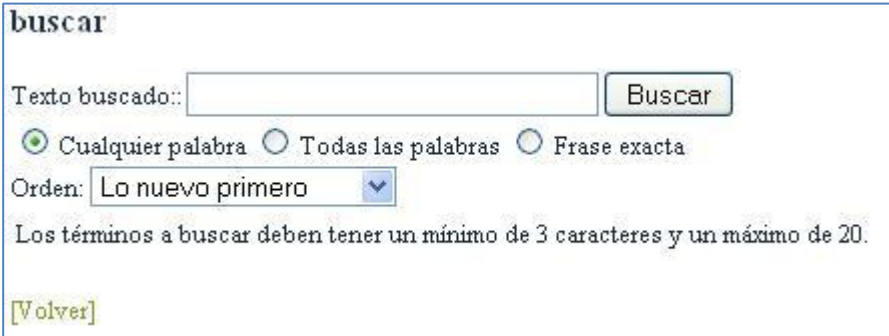
- “Inicio de descargas” (página principal): presenta una lista de las DLL más populares, una lista de los últimos archivos incluidos, un índice de búsqueda por inicial y una explicación de lo que es una DLL, de los problemas que tienen y como se pueden solucionar descargando la DLL correspondiente del sitio.
- “Buscar”: Presenta una búsqueda más completa que los anteriores sitios. Tiene tres modalidades de comparación:
- “Cualquier palabra”: en la que devuelve los archivos que contienen en sus datos al menos una de las palabras introducidas. Más adelante veremos qué datos se almacenan por cada archivo.

- “todas las palabras”: en la que busca los que contengan todas las palabras introducidas.
- “frase exacta”: en la que busca todas las palabras y en el orden introducido.

Presenta cinco formas de ordenar los resultados:

- “Lo nuevo primero”
- “Lo antiguo primero”
- “Lo más leído primero”
- “Alfabéticamente”
- “Sección/Categoría”.

Esta última modalidad presenta los resultados alfabéticamente. No hay diferencia con la modalidad “Alfabéticamente”. Muestra los resultados en una sola página, con lo cual es poco manejable cuando los resultados son muy extensos.



buscar

Texto buscado:

☒ Cualquier palabra ☐ Todas las palabras ☐ Frase exacta

Orden: ▼

Los términos a buscar deben tener un mínimo de 3 caracteres y un máximo de 20.

[\[Volver\]](#)

Ilustración 5- Búsqueda de DLL-download-system

Permite una búsqueda con el nombre del archivo, la descripción, la versión, el copyright y el tamaño, pero no todos los archivos poseen todos los campos. Presenta los resultados en una sola página, en vez de en varias, lo que resulta poco manejable cuando los resultados son muy extensos.



Ilustración 6- Pagina de resultados de DLL-download-system

- “Contactar”: En donde se puede mandar una consulta. No podemos subir archivos.
- “Foro”: como en cualquier foro si queremos abrir una consulta o contestar a una que hubiera nos tenemos que registrar. Tiene una búsqueda en la que se puede buscar por los mensajes o por los usuarios.
- “FAQs”: que contiene preguntas frecuentemente formuladas

Al descargar un archivo nos presenta la información siguiente: nombre del fichero, tamaño y el número de descargas de ese archivo. No presenta información como la descripción, la versión y el copyright. Para ver dicha información tenemos que descargar el archivo y ver sus propiedades.

Los archivos descargados son de tipo ZIP e incluyen la DLL, un documento de ayuda con ayuda y un enlace directo a este sitio.

Podemos subir comentarios por cada archivo. Cada uno tiene una lista de comentarios de los usuarios.

2.5 Repositorio “DLL-Files”



Ilustración 7- RepositorioDLL-Files, <http://www.dll-files.com/>

Las secciones más importantes son las siguientes:

- “Home”: Nos aparece una búsqueda y una lista de iniciales para buscar por la inicial del nombre del archivo. En la búsqueda se puede buscar por nombre del fichero, descripción y versión, se pueden poner varias palabras pero entonces busca la frase exacta, no se pueden buscar varias palabras si no están juntas. Presenta los resultados en una sola página.
- “Download DLL files”: nos aparece otra vez el recuadro de búsqueda y la lista de iniciales.
- “Install DLL files”: Donde nos indican que los archivos que nos descargamos son ZIP y que tenemos que hacer con ellos para reparar un problema de DLL.

- “Help”: Es una sección con la respuesta a preguntas frecuentes.
- “Forum”: foro organizado por temas todos ellos relacionados con DLL.
- “Newsletter”: en la que nos podemos suscribir para que los administradores nos manden información con la que nos podemos beneficiar.
- “Suggest a file”: donde podemos sugerir que incluyan un archivo pero no podemos subirlo nosotros.

Al descargar un archivo nos presenta la siguiente información del mismo: descripción, versión y tamaño. No nos muestra por ejemplo el copyright del archivo. Tendremos que descargarlo para ver el resto de sus propiedades. Presenta los resultados en una sola página.

Searching for: ole
22 files found (sorted by filename)
Make a new search:

asycfilt.dll
Microsoft OLE 2.30 for Windows NT(TM) and Windows
</dllindex/dll-files.shtml?asycfilt>

cygwin1.dll
Cygwin® POSIX Emulation DLL, v. 1001.8.0.0,
[Download Sourcecode](#).
Notice to users:
Because of the special technical nature of the
Cygwin™ DLL in its role providing POSIX-like services
on MS Windows s...
</dllindex/dll-files.shtml?cygwin1>

fxscIntr.dll
Fax Console Resources, v. 5.2.1776.0
</dllindex/dll-files.shtml?fxscIntr>

mfcans32.dll
OLE2ANSI Library - Retail Version, v. 3.2.1.0
</dllindex/dll-files.shtml?mfcans32>

Suggest us to add a dll-file to our website:

Filename: *

Ilustración 8- Pagina de resultados de DLL-files

Los archivos a descargar son comprimidos de tipo ZIP e incluyen la DLL y un documento de ayuda. Se pueden subir comentarios a cada archivo. En algunos de los archivos nos permite descargar el código fuente de la DLL.

2.6 Repositorio “A Free DLL”

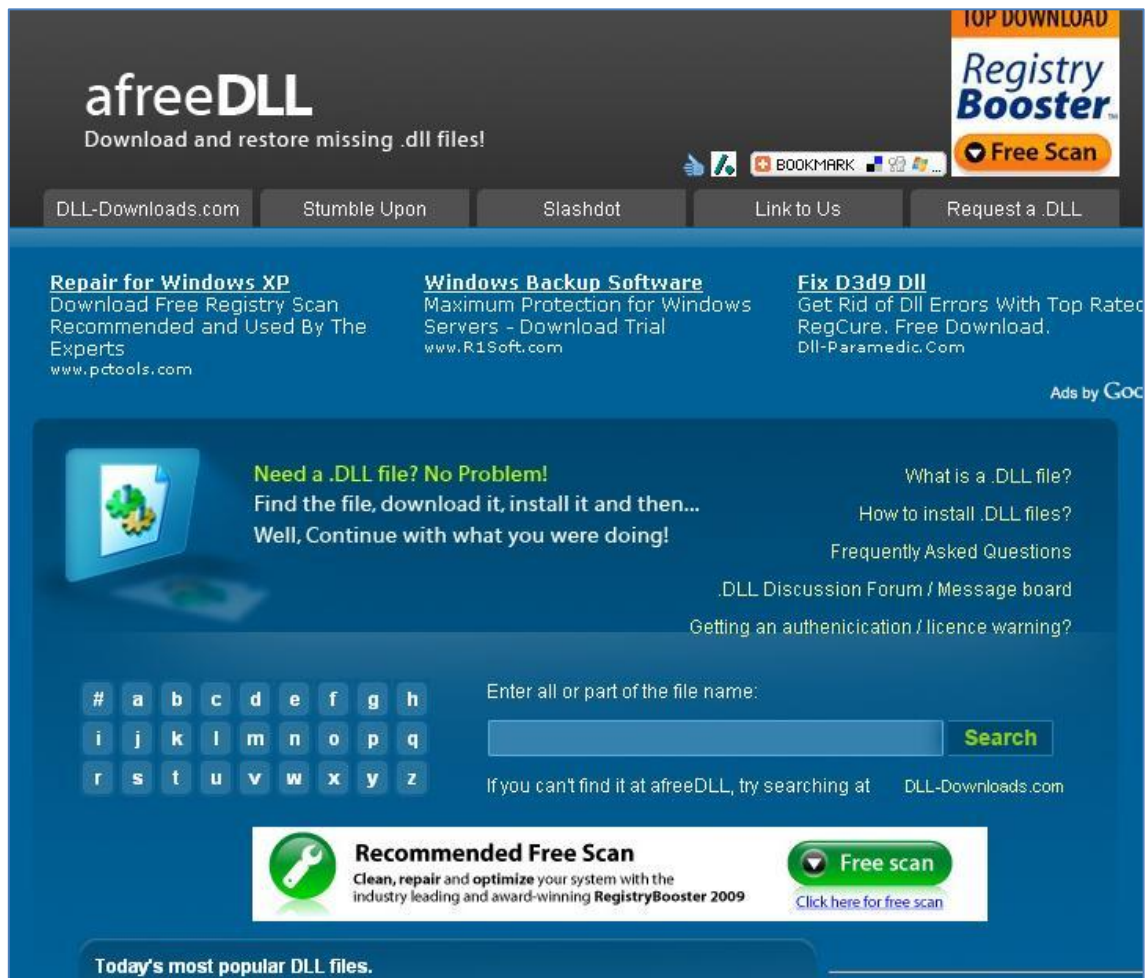
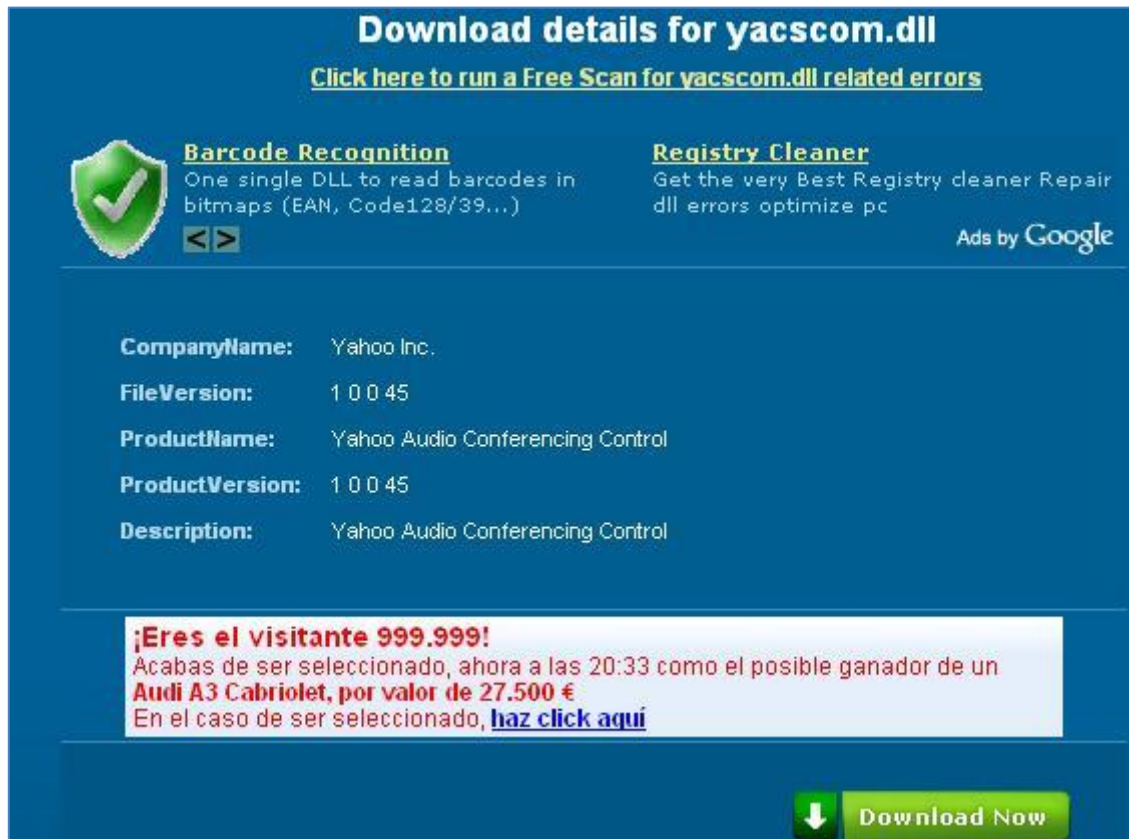


Ilustración 9- Repositorio AfreeDLL, <http://www.afreedll.com/>

La página principal tiene un recuadro de búsqueda y una lista de iniciales. En la búsqueda solo busca por el nombre del fichero, aunque podemos poner solo una parte. Los resultados los muestra en una sola página.

Al ir a bajarnos uno de ellos nos dan bastante información del archivo: Nombre de la compañía (copyright), versión del archivo, nombre del producto, versión del producto, descripción, aunque no podemos buscar por estos datos.



Download details for yacscm.dll

[Click here to run a Free Scan for yacscm.dll related errors](#)

Barcode Recognition
One single DLL to read barcodes in bitmaps (EAN, Code128/39...)

Registry Cleaner
Get the very Best Registry cleaner Repair dll errors optimize pc

Ads by Google

CompanyName:	Yahoo Inc.
FileVersion:	1.0.0.45
ProductName:	Yahoo Audio Conferencing Control
ProductVersion:	1.0.0.45
Description:	Yahoo Audio Conferencing Control

¡Eres el visitante 999.999!
Acabas de ser seleccionado, ahora a las 20:33 como el posible ganador de un Audi A3 Cabriolet, por valor de 27.500 €
En el caso de ser seleccionado, [haz click aquí](#)


 **Download Now**

Ilustración 10- Información de archivo de AfreeDLL

Nos aparecen varios enlaces más:

- [“What is a .DLL file”](#): donde nos explican lo que es y cómo funciona una DLL.
- [“How to install DLL files”](#): Donde nos dicen donde los tenemos que instalar.
- [“Frequently asked questions”](#): Contestación a las preguntas más frecuentes.
- [“DLL discussion forum/Message Board”](#): foro en el que podemos mandar mensajes sin necesidad de registrarnos. Podemos registrarnos si queremos.
- [“Request a DLL”](#): donde podemos sugerir que incluyan una DLL. Pero no podemos subirla nosotros.

2.7 Repositorio “DLL-Downloads”



Ilustración 11- Repositorio Dll-downloads, <http://www.dll-downloads.com>

En este sitio podemos descargarnos ficheros de tipo DLL, OCX y DRV.

Las secciones más importantes de este repositorio son las siguientes:

- **“Home”**: donde nos explican lo que son los ficheros DLL, OCX y DRV, para que sirvan como se usan y donde se suelen ubicar. También nos aparece una lista de Sistemas Operativos para descargar archivos de cada sistema.
- **“Download a File”**: donde nos encontramos enlaces para descargarnos archivos de cada uno de los tipos: DLL, DRV y OCX. Cuando pulsamos en cada uno de ellos nos aparecen una lista de enlaces con todos los sistemas operativos para descargar archivos para cada sistema operativo. No nos aparece Windows Vista.
 - Dentro de esta sección también nos encontramos con el enlace **“Comonly Requested Files”** en el que si pulsamos nos sale una lista bastante extensa y poco manejable de archivos frecuentemente descargados.
 - También en esta sección encontramos el enlace **“Search for a DLL, DRV or OCX file”** para buscar archivos. La búsqueda es bastante limitada, solo

podemos buscar por nombre y este tiene que ser exacto. No podemos poner solo una parte del mismo. En una lista desplegable al lado del recuadro de búsqueda tenemos que indicarle el tipo de archivo que queremos descargar (DLL, DRV u OCX). Tenemos que especificar uno de los tipos obligatoriamente, no se puede dejar en blanco para que busque en cualquiera de ellos. Tampoco tenemos posibilidad de buscar por inicial, como en los otros sitios vistos anteriormente.

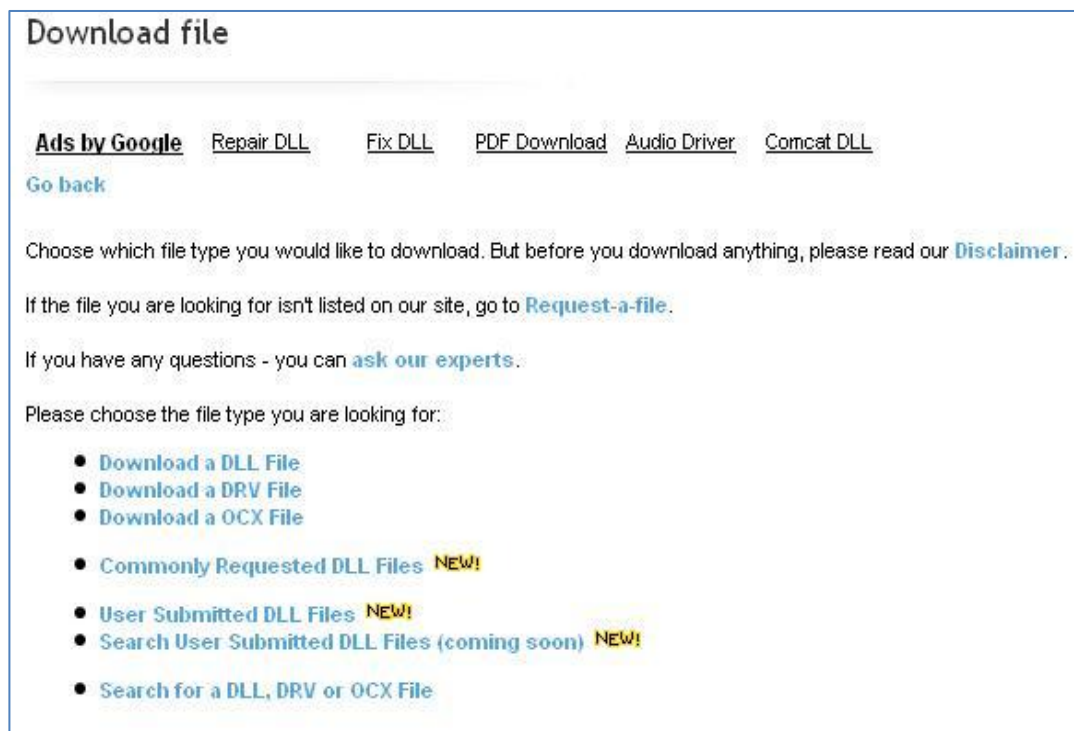


Ilustración 12- Página para descargar Archivo de DLL-downloads

Download a DLL file

[Go back](#)

Choose which dll-file you would like to download. But before you download anything, please read our [Disclaimer](#).

If the file you are looking for isn't listed below, go to [Request-a-file](#).

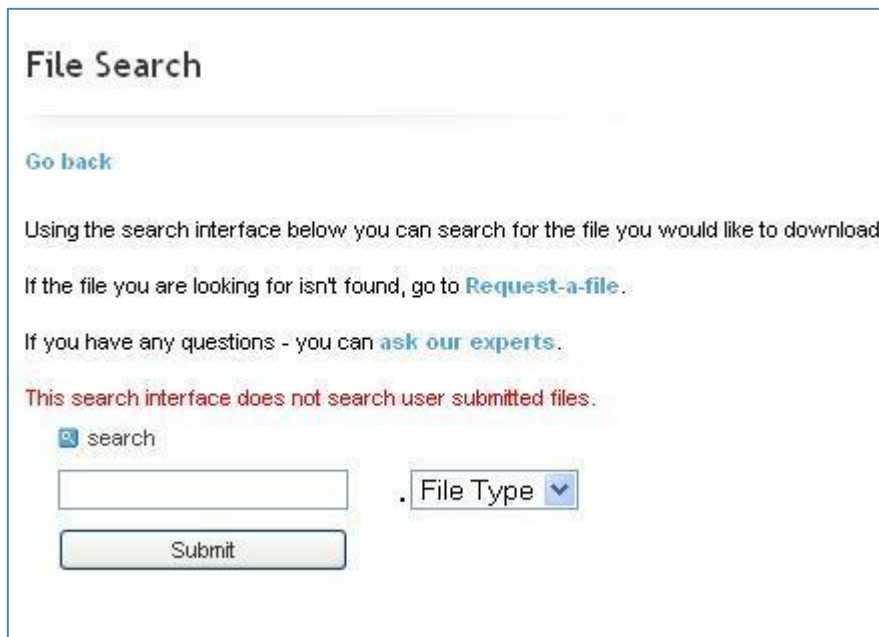
If you have any questions - you can [ask our experts](#).

Please choose your current operating system:

Download Commonly Requested DLL Files **NEW!**

- [Download a DLL File for Windows 3.x](#)
- [Download a DLL File for Windows 95](#)
- [Download a DLL File for Windows 98](#)
- [Download a DLL File for Windows 98 Second Ed.](#)
- [Download a DLL File for Windows ME](#)
- [Download a DLL File for Windows 2000 Professional](#)
- [Download a DLL File for Windows 2000 Server](#)
- [Download a DLL File for Windows 2000 Advance Server](#)
- [Download a DLL File for Windows 2000 Datacenter Server](#)
- [Download a DLL File for Windows XP Home](#)
- [Download a DLL File for Windows XP Professional](#)
- [Download a DLL File for Windows 2003 Standard Ed.](#)
- [Download a DLL File for Windows 2003 Web Ed.](#)
- [Download a DLL File for Windows 2003 Small Business Ed.](#)

Ilustración 13- Pagina para elegir sistema operativo en DLL-downloads



File Search

[Go back](#)

Using the search interface below you can search for the file you would like to download.

If the file you are looking for isn't found, go to [Request-a-file](#).

If you have any questions - you can [ask our experts](#).

This search interface does not search user submitted files.

search

. File Type

Ilustración 14- Pagina de búsqueda en DLL-downloads

- “How to install a DLL file”: donde nos dicen los pasos necesarios para instalar un archivo DLL en Windows.
- “Live help”: es un servicio de soporte 24 x 7 (a cualquier hora, cualquier día de la semana) de pago.
- “Request a missing file”: donde podemos solicitar un archivo para que lo añadan a la base de datos. Por el momento no se pueden subir archivos aunque tienen previsto añadirlo.

Al descargar un archivo no nos presenta información del mismo. Lo tenemos que descargar para enterarnos.

Los archivos descargados son comprimidos de tipo ZIP y contienen la DLL y un documento de ayuda con notas de instalación.

2.8 Comparación de los repositorios

Después de visitar todos estos sitios vemos que unos tienen unas funcionalidades y otros otras pero ninguno reúne todas. Por otra parte vemos que hay funcionalidades que ninguno tiene. A continuación vamos a hacer un resumen comparativo con respecto a estas funcionalidades. Se van a considerar las siguientes:

- lista de iniciales
- búsqueda por nombre de archivo
- búsqueda por una parte del nombre del archivo
- búsqueda por varias propiedades
- resultados paginados
- presentación de información del archivo
- lista de archivos frecuentemente descargados
- lista de archivos recientemente incluidos
- clasificación por Sistema Operativo
- otros archivos además de DLL
- subir comentarios a cada archivo
- contacto / sugerir archivos
- subir nuestros archivos
- foro

A continuación presentamos una tabla comparativa de los repositorios de DLL en torno a estas funcionalidades:

Repositorio

Requisito	Publispain	My-DLL	Archivos-DLL	DLL-Download-System	DLL-Files	A Free DLL	DLL-Downloads
Buscar por inicial	SI	NO	SI	SI	SI	SI	NO
búsqueda por nombre de archivo	NO	SI	SI	SI	SI	SI	SI
búsqueda por una parte del nombre	NO	NO	SI	SI	SI	SI	NO
búsqueda por varias propiedades	NO	NO	NO	SI	SI	NO	NO
Resultados paginados	NO	NO	NO	NO	NO	NO	NO
Presentación de alguna información del archivo	NO	NO	SI	SI	SI	SI	NO
Lista de archivos frecuentemente descargados	NO	NO	SI	SI	NO	NO	SI
Lista de archivos recientemente incluidos	NO	NO	NO	SI	NO	NO	NO
Clasificación por Sistema Operativo	NO	NO	NO	NO	NO	NO	SI
Otros archivos además de DLL	NO	NO	NO	NO	NO	NO	SI
Subir comentarios a cada archivo	NO	NO	SI	SI	SI	NO	NO
Contacto / Sugerir archivos	NO	SI	SI	SI	SI	SI	SI
Subir nuestros archivos	NO	NO	NO	NO	NO	NO	NO
Foro	NO	NO	NO	SI	SI	SI	NO

Tabla 1- Comparativa de los repositorios de archivos dll

Vemos que en cuanto a la búsqueda solo dos sitios ofrecen una búsqueda facetada, es decir por varias propiedades. Son DLL-download-system y DLL-files. DLL download system es el más avanzado en este aspecto ya que propone una búsqueda por nombre, descripción, versión, copyright y tamaño. DLL-files propone una búsqueda solo por nombre, descripción y versión. El resto de los sitios tienen una búsqueda muy limitada si es que existe, donde solo podemos buscar por nombre y en varios tendremos que poner el nombre exacto junto con la extensión del archivo, la cual se puede obviar al no tener varios tipos de archivo.

En cuanto a información relacionada con el archivo no todos los sitios la proporcionan. El sitio más completo es “afreeDLL” que incluye: nombre, descripción, versión, copyright, nombre del producto y versión del producto. Sin embargo afreeDLL no tiene búsqueda facetada, por ejemplo.

La lista de archivos frecuentemente descargados no está presente en todos los sitios vistos, lo mismo pasa con subir comentarios a cada archivo para ver la fiabilidad del mismo, y tampoco está presente en todos la posibilidad de sugerir archivos y el foro.

La lista de archivos recientemente incluidos solo la presenta DLL-download-system.

La posibilidad de incluir otros archivos como OCX y DRV solo la tiene DLL-downloads. Sin embargo este sitio tiene una búsqueda muy limitada.

Por otra parte vemos que no hay ninguno que ofrezca añadir archivos por parte de cualquier usuario que visite el sitio. Tampoco vemos que alguien ofrezca los resultados en varias páginas, lo cual resultaría más manejable cuando los resultados son amplios. En la búsqueda por inicial tampoco estaría de más paginar los resultados.

3. Marco del proyecto

En el marco del proyecto hablaremos la metodología y el ciclo de vida utilizados, así como de la adaptación de los mismos a las necesidades de nuestro sistema, dando el contenido concreto en nuestro caso de cada fase de dicho ciclo de vida.

3.1 *Paradigma de desarrollo utilizado*

En el proyecto Repositorio de Archivos DLL se va a seguir el paradigma de Orientación a Objetos debido a que presenta las siguientes ventajas frente al paradigma procedimental o tradicional:

- Las personas piensan en términos nominales (objetos) y verbales (comportamientos de los objetos). Por tanto la orientación a objetos es más próxima a la realidad y ello conlleva mayor entendimiento entre los clientes y los encargados de desarrollar el sistema.
- En cuanto a la facilidad de modificación, el software desarrollado mediante la orientación a objetos es más robusto y más fácil de modificar.
- El software orientado a objetos presenta mayor facilidad de reutilización mediante la herencia de clases y la instanciación de objetos.
- Gracias al polimorfismo en la orientación a objetos se facilita la configuración de casos especiales.
- Existe mayor facilidad a la hora de probar el software orientado a objetos al estar modularizado por casos de uso del sistema [7].

3.2 *Metodología de desarrollo utilizada*

En el proyecto Repositorio de Archivos DLL se va a utilizar la metodología del Proceso Unificado de Desarrollo de Software o simplemente Proceso Unificado (en inglés Unified Process - UP) debido a que contempla varias iteraciones. En una primera iteración se incorporaran las funcionalidades básicas y en sucesivas iteraciones se añadirán funcionalidades

más superfluas en función del tiempo disponible. No nos conviene por ejemplo la metodología en cascada, la cual es apropiada para proyectos cuyos requisitos no van a cambiar, puesto que no contempla más de una iteración.

3.2.1 Adaptación del Proceso Unificado a las necesidades del producto

UP es un proceso genérico de desarrollo de software que se tiene que personalizar para cada proyecto en particular porque todos los proyectos de software tienden a ser diferentes [8].

Las peculiaridades de nuestro proyecto son las siguientes:

- En las charlas mantenidas con el tutor se han determinado todos los requisitos, y de ellos cuáles son obligatorios y cuáles opcionales. Es poco probable que aparezcan requisitos nuevos.
- No sabemos cuántos requisitos se van a implementar por razones de tiempo. Por tanto se opta por realizar varias iteraciones yendo de más importancia a menos.

Para poder definir las fases e iteraciones de la adaptación del proceso, vamos a determinar las categorías en que se van a clasificar los requisitos y los casos de uso que los comprenden en función de la importancia.

De tal manera, se van a clasificar los casos de uso en tres categorías. Las categorías son:

- Crítico: el caso de uso debe ser implementado obligatoriamente, de lo contrario el sistema no cumpliría lo acordado con el cliente, en este caso el tutor del proyecto.
- Importante: el caso de uso se podría omitir pero eso afectaría la usabilidad del sistema y la satisfacción del cliente.
- Útil: el caso de uso se podría omitir sin impacto en la aceptación del sistema por parte del cliente [9].

Para los requisitos concretos se hace una segunda clasificación con los valores *Obligatorio* y *Opcional*. Dentro de un caso de uso se van a considerar requisitos obligatorios aquellos que son esenciales para el caso de uso. Serán opcionales aquellos que se pueden omitir sin impacto en la satisfacción del cliente.

Esta clasificación nos da la posibilidad de hacer varias iteraciones en cada una de las cuales se puede implementar un conjunto de requisitos clasificados dentro de una combinación de ambos criterios, es decir:

Iteración 1: requisitos obligatorios de casos de uso *críticos*

Iteración 2: requisitos obligatorios de casos de uso importantes

Iteración 3: requisitos obligatorios de casos de uso útiles

Iteración 4: requisitos opcionales de casos de uso críticos

Iteración 5: requisitos opcionales de casos de uso importantes

Iteración 6: requisitos opcionales de casos de uso útiles.

Una vez definidos los tipos de prioridad pasamos a describir las fases de UP adaptadas a nuestro proyecto.

Fase de Comienzo

En nuestro caso consistirá únicamente en dar una visión general del proyecto. Pensamos que para este proyecto no es aplicable realizar un estudio de viabilidad ni un prototipo. En cuanto al plan de proyecto inicial nos parece que tenemos muy pocos detalles y sería muy inexacto. Por otra parte no se nos exige.

Por tanto esta fase solo tendrá una iteración y solo contará con un solo flujo de trabajo que es el de Requisitos.

Fase de Elaboración

Debido a lo que hemos comentado que previamente se han determinado todos los requisitos y su importancia pensamos que en el flujo de requisitos lo correcto es incluir la especificación de requisitos completa determinando todos los casos de uso clasificándolos por las categorías comentadas y dentro de ellos incluir todos los requisitos clasificados a su vez por

las suyas. Dejaremos para la fase de Construcción únicamente retoques ante alguna omisión o incoherencia.

Para el resto de flujos lo ideal es dar tratamiento a los requisitos **obligatorios** de casos de uso **críticos** ya que en esta fase se suele crear la línea base del producto. Por tanto se realizara el análisis el diseño y la arquitectura en UML de dichos requisitos y se creara el esqueleto de la aplicación mediante generación automática de código.

Esta fase contemplará una sola iteración donde tendrán cabida todos los flujos de trabajo.

Fase de Construcción

En ella como hemos dicho apenas tendrá cabida el flujo de requisitos.

Se contemplan para esta fase varias iteraciones en las que todos los flujos restantes son importantes. Dichas iteraciones resultan de las combinaciones de los criterios de importancia de los requisitos.

Iteración 3: requisitos **obligatorios** de casos de uso **importantes**

Iteración 4: requisitos **obligatorios** de casos de uso **útiles**

Iteración 5: requisitos **opcionales** de casos de uso **críticos**

Iteración 6: requisitos **opcionales** de casos de uso **importantes**

Iteración 7: requisitos **opcionales** de casos de uso **útiles**

Fase de Transición

Se contempla para esta fase la puesta en producción y las pruebas de carga.

Tabla con el Resultado

Todo lo dicho anteriormente se plasma en una tabla que proporcionará el marco del desarrollo de este producto.

Fases del proceso

Flujos de trabajo	Fase de Comienzo	Fase de Elaboración	Fase de construcción				Fase de Transición
Requisitos	Visión General	ERS completa	Retoques requisitos	Retoques	...	Retoques	No aplicable
Análisis	No aplicado	Análisis req. Cr/Ob	Retoques análisis req. Cr/Ob	Analisis requisitos Im/Ob	...	Análisis Requisitos Ut/Op	No Aplicable
Diseño y arquitectura	No aplicado	Diseño req. Cr/Ob Arq. req. Cr/Ob	Retoques diseño y arquitectura req. Cr/Ob	Diseño y arquitectura req. Im/Ob	...	Diseño y arq. req. Ut/Op	Retoques
Métrica del producto	No aplicado	Estimación y planificación req. Cr/Ob	Ajustes planificación	Ajustes planificación		Ajustes Planificación	Retoques
Implementación	No aplicado	Esqueleto aplicación. Generación código	Implementación requisitos Cr/Ob	Implementación requisitos Im/Ob	...	Implementación requisitos Ut/Op	Retoques
Prueba	No aplicable	Pruebas línea base arquitectura	Prueba requisitos Cr/Ob	Prueba requisitos Im/Ob	...	Prueba requisitos Ut/Op	Pruebas de carga
Iteraciones	Preliminares	It. 1: análisis y diseño requisitos obligatorios de casos de uso críticos. Línea base arquitectura	It. 2: implementación requisitos obligatorios de casos de uso críticos	It.3: implementación requisitos obligatorios de casos de uso importantes	...	It.7 : implementación requisitos opcionales de casos de uso útiles	It. 8

Tabla 2- Adaptación del Proceso Unificado a nuestro producto

3.2.2 El Proceso Unificado y la metodología SunTone

Pensamos que el proceso Unificado deja muchos detalles sin definir dentro de los flujos de trabajo. Es por eso que dentro del marco de la metodología de UP se van a seguir las directrices proporcionadas por la multinacional Sun Microsystems Inc. en su metodología SunTone desarrollada a partir del Proceso Unificado y expuesta en sus cursos de certificación. Nos parece una guía más exhaustiva.

4. Fase de Comienzo

En la fase de comienzo daremos una visión general del proyecto y ahondaremos un poco más en los detalles de la solución que ya se introdujeron en capítulos anteriores.

4.1 Vista general del proyecto

Este apartado se ha elaborado a partir de la elaboración del estado del arte y de las entrevistas mantenidas con el tutor del proyecto con el fin de transmitir la idea concreta que él tenía para este producto. El tutor representa la parte interesada o cliente.

4.1.1 Introducción

Se pretende crear un repositorio Web de DLL que permita la descarga de archivos DLL mediante búsqueda facetada y que recoja las funcionalidades de todos los repositorios vistos en el estado de la cuestión y junte lo mejor de cada uno, mejorando la oferta actual. Al igual que los repositorios propuestos será una aplicación Web a la que se pueda acceder desde cualquier parte del mundo. Será una aplicación enfocada tanto a usuarios con problemas de DLL como a programadores que necesiten funcionalidades ya implementadas.

4.1.2 Solución propuesta

El Repositorio DLL permitirá una búsqueda facetada de archivos (como la de *DLL-download-system*). Se añadirán las modalidades de búsqueda de *DLL-download-system*: “cualquier palabra” “todas las palabras” y “frase exacta”, así como sus distintas maneras de ordenar resultados: orden creciente de fecha en que se incluyó el archivo, orden decreciente, primero los más descargados y alfabéticamente por nombre. No se incluirá la opción “sección / categoría”. Se facilitará una búsqueda por inicial. Se proporcionará asimismo la funcionalidad de buscar por Sistema Operativo mediante enlaces (como en *DLL-downloads*) y también en la búsqueda por palabras. Los resultados se mostraran en varias páginas con un enlace por página.

En la información del archivo se recopilara la de todos los repositorios. Se presentara la información de *AfreeDLL* que es el más información facilita (nombre, copyright, versión, nombre del producto, versión del producto, descripción), más el número de descargas de cada archivo y el tamaño, que son facilitados por *DLL-download-system*. Se propone añadir además el resto de detalles de versión que se pueden ver al mostrar las propiedades de una DLL. Estos son por tanto los datos (facetas) por los que se clasificara a los archivos en el sistema:

- Nombre del archivo.
- Número de descargas.
- Tamaño.
- Fecha de subida al sistema.
- Versión del archivo.
- Versión del producto.
- Sistema Operativo.
- Tipo de archivo.
- Subtipo de archivo: función del archivo, solo nos interesa en caso de que el archivo sea Driver o Font.
- Fecha de creación del archivo.
- Organización que ha creado el archivo.
- Descripción de las funciones del archivo.
- Nombre interno del archivo.
- Copyright del archivo.
- Marcas registradas del archivo.
- Nombre original del archivo.
- Nombre del producto en el que está incluido el archivo.

Nuestro repositorio facilitará una lista de archivos frecuentemente descargados, una lista de archivos recientemente incluidos, la funcionalidad de subir comentarios a cada archivo, un foro de consulta y el almacenamiento de otros archivos además de DLL. Se almacenaran además de DLL, archivos OCX y DRV, como en DLL-downloads. Se añade el tipo de archivo FON que ninguno lo incluye.

Por último se ofrecerá la funcionalidad de subir archivos por parte del cualquier usuario, posibilidad que no ofrece ninguno de los sitios estudiados.

4.1.3 Riesgos

El único riesgo encontrado está relacionado con las habilidades y es la falta de conocimientos y experiencia en lenguajes de programación orientada a objetos.

4.1.4 Restricciones

No se contempla ninguna restricción ya que no se nos impone ninguna plataforma específica ni ninguna tecnología ni tampoco el proyecto tiene un plazo de entrega inflexible. Tampoco debe interactuar con ningún otro sistema externo existente. El Repositorio de DLL es un sistema independiente por completo.

5. Fase de Elaboración

En la fase de elaboración se incluyen las fases de requisitos, análisis, diseño, arquitectura, implementación e implantación de nuestro producto, o lo que es lo mismo la primera iteración del ciclo de vida escogido, el ciclo de vida en espiral.

5.1 *Flujo de Requisitos*

En el flujo de requisitos incluimos la especificación formal de requisitos de software (ERS) para este sistema según el estándar IEE-830.

5.1.1 Introducción

Este documento es una Especificación de requisitos de software (ERS) para el sistema de Repositorio Web de DLL. Su contenido ha sido elaborado en colaboración con el tutor del proyecto.

Propósito

El objetivo de esta especificación es definir de forma clara y precisa las funcionalidades del sistema *Repositorio Web de DLL*. Servirá de guía para el diseño y desarrollo del sistema. Igualmente está destinado a la parte interesada y a los posibles usuarios finales.

Ámbito del sistema

Los motivos que impulsan la realización del Sistema Repositorio Web de DLL son la deficiente oferta actual del mercado en cuanto a aplicaciones similares se refiere, donde consideramos que se pueden añadir nuevas funcionalidades así como mejoras en usabilidad.

Pensamos que tanto un usuario que tiene un problema con alguna DLL, como un programador realizando su aplicación, o cualquier empresa de desarrollo de software que pretenda reutilizar sus librerías de software podrían beneficiarse de este producto.

Contexto o Perspectiva del sistema

No está previsto que el Repositorio Web de DLL interactúe con otros sistemas.

Parte interesada

La parte interesada (en inglés StakeHolders) en el proyecto es la Universidad Carlos III de Madrid a través del tutor Jorge Morato Lara.

Acrónimos y Abreviaturas

DLL – Dinamic Link Library , Librería de enlace dinámico.

ERS – Especificación de Requisitos de Software.

SO – Sistema Operativo.

SSOO – Sistemas Operativos.

TF – Texto del Foro, consulta enviada al foro o respuesta a alguna consulta.

TTFF – Textos del Foro.

Vista general del documento

El documento consta de seis secciones:

1. **Introducción:** donde se proporciona una visión general del documento ERS.
2. **Restricciones y Suposiciones,** donde se describen aquellas limitaciones que se imponen sobre los desarrolladores del producto como pueden ser políticas de empresa, limitaciones de hardware, interfaces con otras aplicaciones, operaciones paralelas, funciones de auditoria, funciones de control o lenguajes de programación.

Por otra parte se describen los factores que en caso de cambiar pueden afectar a los requisitos.

3. **Riesgos**, donde se describe todo aquello que puede provocar que el proyecto fracase.
4. **Requisitos funcionales**, donde se muestra un resumen de las funciones principales del producto. Por otra parte se describen las características generales de los usuarios, incluyendo nivel educacional, experiencia, y experiencia técnica. Se describen asimismo los casos de uso del sistema, las aplicaciones que forman parte del producto y por último se describen los requerimientos funcionales a un nivel de detalle suficiente como para permitir realizar un diseño que satisfaga dichos requisitos, y que permita realizar una planificación de pruebas que demuestre que el producto cumple con los requisitos. Además de describe la relación entre requisitos y casos de uso del sistema. Todos los requisitos están englobados en el caso de uso que los contiene.
5. **Requerimientos no Funcionales**, donde se describen aquellos requerimientos relacionados con las calidades del producto, que son principalmente las relativas a **rendimiento, usabilidad y seguridad**.
6. **Glosario del proyecto**, donde se describen los términos del dominio de la aplicación, sinónimos, términos tecnológicos y términos de desarrollo de software.

Referencias

En el apartado restricciones así como el de riesgos se hace referencia al documento Visión General del Proyecto incluido en esta memoria.

5.1.2 Restricciones y Suposiciones

Como ya hemos comentado en el apartado Visión General del Proyecto, la parte interesada, a través del tutor del proyecto, no nos impone ninguna restricción ni relacionada con la plataforma ni con la tecnología ni con el lenguaje de programación y tampoco nos impone un plazo de entrega inflexible, ni debe interactuar con ningún otro sistema externo existente.

5.1.3 Riesgos

Como hemos comentado en el apartado Visión General del proyecto, El único riesgo encontrado está relacionado con las habilidades y es la falta de conocimientos y experiencia en lenguajes de programación orientada a objetos.

5.1.4 Restricciones

No se contempla ninguna restricción ya que no se nos impone ninguna plataforma específica ni ninguna tecnología ni tampoco el proyecto tiene un plazo de entrega inflexible. Tampoco debe interactuar con ningún otro sistema externo existente. El Repositorio de DLL es un sistema independiente por completo.

5.1.5 Requisitos funcionales

Funciones principales del producto

Descarga de Archivos

El Repositorio de DLL permitirá que cualquier usuario se descargue los archivos que quiera. Para ello primero debe hacer una búsqueda o bien descargar uno de los archivos más descargados o bien uno de los últimos incluidos. Para hacer dispone de dos opciones:

- Búsqueda facetada de archivos: El Repositorio DLL permitirá una búsqueda facetada de archivos, donde se buscara por cualquiera de las propiedades del archivo.
- Búsqueda no refinada: El Repositorio de DLL facilitara una búsqueda por inicial del nombre del archivo así como una búsqueda por Sistema Operativo y otra por tipo de archivo. También presentara una lista de archivos frecuentemente descargados y una lista de archivos recientemente incluidos.

Subida de archivos

El Repositorio de DLL ofrecerá la posibilidad de subir archivos por parte del cualquier usuario.

Foro de consulta

El Repositorio de DLL ofrecerá un foro de consulta para que cualquiera pueda enviar su consulta o contestar a la consulta de otro usuario.

Sugerencias de los usuarios

El Repositorio de DLL ofrecerá la posibilidad de que cualquier usuario pueda sugerir que se incluya algún archivo en el repositorio o cualquier otra sugerencia.

Comentarios a los archivos

El Repositorio de DLL ofrecerá la posibilidad de enviar comentarios a los archivos por parte del cualquier usuario.

Características de los usuarios

El sistema está pensado para cualquier persona que sepa manejar un ordenador. Se pretende que el sistema ofrezca una interfaz de usuario intuitiva y fácil de manejar en poco tiempo.

Casos de Uso

Siguiendo las directrices de Sun Microsystems Inc., hemos incluido el diagrama de casos de uso en esta sección ya que proporciona las siguientes ventajas:

- El diagrama de caso de uso proporciona una representación visual del apartado de Requerimientos Funcionales de la ERS.
- Incluir el diagrama de Caso de Uso dentro de la ERS facilita mantener los dos documentos sincronizados.
- El propósito principal del diagrama de caso de uso es proporcionar una vista general del comportamiento del sistema para el cliente [10].

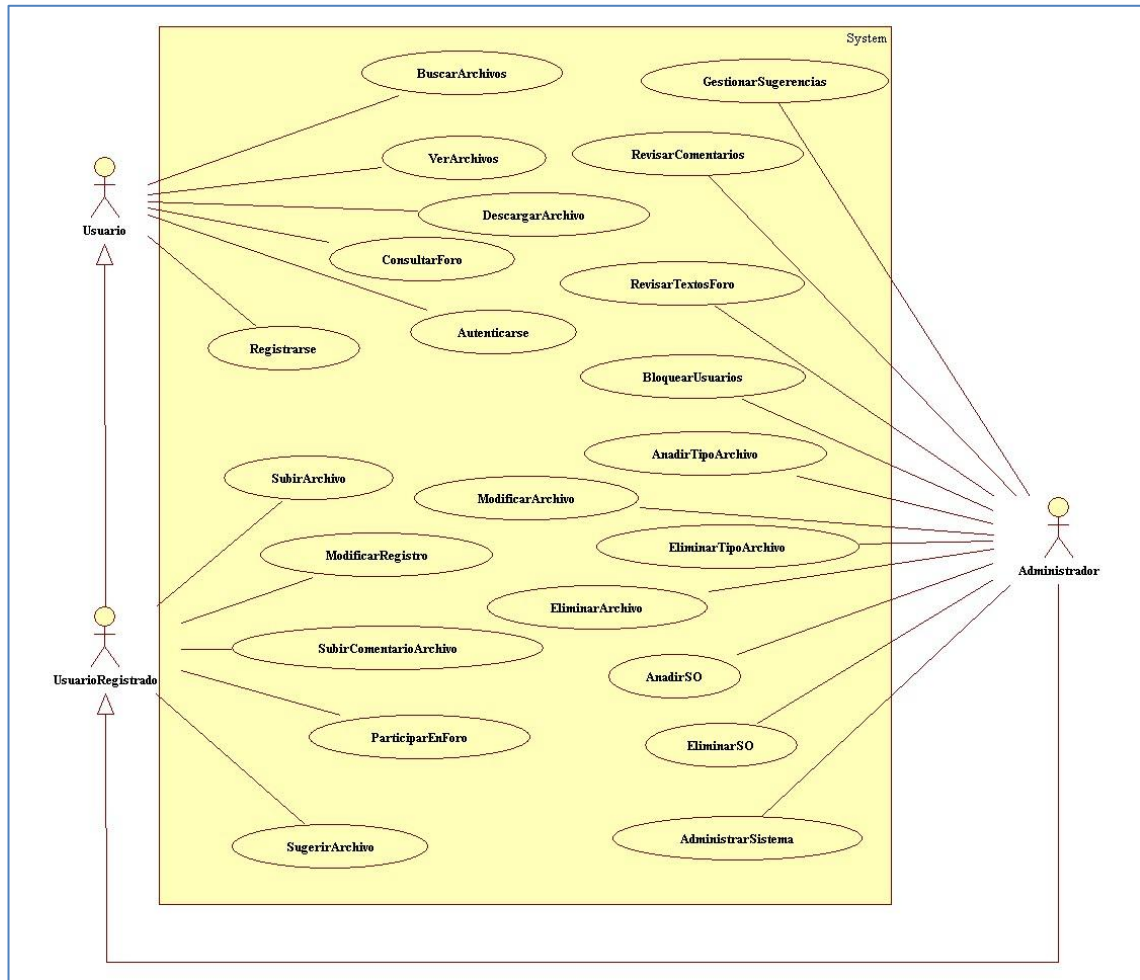


Ilustración 15- Diagrama de casos de uso del sistema

El caso de uso *AdministrarSistema* correspondería a una serie de actividades que habría que realizar si el sistema estuviera en explotación como por ejemplo supervisar la seguridad del sistema ante ataques o la actualización de los contenidos del sitio. No se va a dar tratamiento a este caso de uso.

Los casos de uso son los siguientes:

Tabla 3- Casos de uso del sistema

Nombre del caso de uso	Prioridad	ID	Descripción del caso de uso
BuscarArchivos	Crítico	C1	Un usuario hace una búsqueda de archivos de cualquiera de las posibles formas: búsqueda por criterios, listado de archivos cuyo nombre empiece por determinada inicial, listado de archivos de un Sistema Operativo determinado o bien listado de archivos de un determinado tipo.
VerArchivos	Crítico	C2	Un usuario solicita ver los detalles de un archivo.
DescargarArchivo	Crítico	C3	Un usuario solicita la descarga de un archivo.
Registrarse	Crítico	C4	Un usuario rellena sus datos de registro y los envía para que el sistema añada al nuevo usuario.
ModificarRegistro	Crítico	C5	Un usuario registrado solicita modificar sus datos de registro.
Autenticarse	Crítico	C6	Un usuario registrado introduce su login y su contraseña para poder utilizar las funcionalidades de usuario registrado o de administrador.
SubirArchivo	Crítico	C7	Un usuario registrado sube una copia de un archivo desde su equipo al repositorio, para que el sistema lo incluya.
EliminarArchivo	Crítico	C8	Un usuario administrador borra un archivo.
ModificarArchivo	Crítico	C9	Un usuario administrador modifica las propiedades de un archivo.
SubirComentarioArchivo	Crítico	C10	Un usuario registrado sube un comentario relacionado con un archivo del repositorio.

Nombre del caso de uso	Prioridad	ID	Descripción del caso de uso
RevisarComentariosArchivo	Importante	I1	Un administrador solicita un listado en pantalla para revisión de comentarios a archivo ya sean pendientes de revisión como ya revisados. El administrador da el visto bueno o elimina uno o varios comentarios a archivo.
ConsultarForo	Importante	I2	Un usuario solicita ver el foro. El usuario solicita consultar la información detallada de algún tema del foro.
ParticiparEnForo	Importante	I3	Un usuario participa activamente en el foro ya sea mandando una consulta o contestando a otra consulta existente.
RevisarTextosForo	Importante	I4	Un administrador solicita un listado en pantalla para revisión de los textos del foro (consultas o respuestas) ya sean pendientes de revisión o no. El administrador da el visto bueno a uno o varios textos del foro o elimina uno o varios textos del foro.
SugerirArchivo	Importante	I5	Un usuario envía una petición sugiriendo que se incluya un archivo en el repositorio o cualquier otra sugerencia.
GestionarSugerencias	Importante	I6	Un administrador solicita un listado en pantalla de las sugerencias de los usuarios. Si lo cree oportuno solicita eliminar una o varias sugerencias.
BloquearUsuarios	Útil	U1	Un administrador solicita un listado de usuarios. El administrador solicita consultar la información detallada de un usuario. El administrador solicita bloquear un usuario pernicioso para el sistema.
AnadirTipoArchivo	Útil	U2	Un administrador añade un nuevo tipo de archivo a los existentes en el repositorio.

Nombre del caso de uso	Prioridad	ID	Descripción del caso de uso
EliminarTipoArchivo	Útil	U3	Un administrador elimina un tipo de archivo existente en el repositorio y con el todos los archivos de ese tipo.
AnadirSO	Útil	U4	Un administrador añade un nuevo sistema operativo en el repositorio.
EliminarSO	Útil	U5	Un administrador elimina un sistema operativo existente en el repositorio y con el todos los archivos de ese sistema operativo.

Aplicaciones

El producto consta de una sola aplicación Web, denominada RepositorioDLL, que soporta todos los casos de uso.

Requisitos Detallados

En esta sección se incluye una lista completa de los requisitos funcionales detallados que comprende cada caso de uso. Para ello se les da un identificador que comienza por el identificador del caso de uso al que se le suma el identificador de requisito. Se hace por tanto una clasificación de la prioridad de los requisitos con dos términos, el primero es relativo al caso de uso que contiene al requisito y el segundo término es relativo al requisito en sí. Por ejemplo *crítico/obligatorio* alude a un requisito *obligatorio* perteneciente a un caso de uso *crítico*.

Los requisitos obligatorios se describirán con el termino ***debe*** y los requisitos opcionales con el termino ***podría***.

Los requisitos funcionales detallados son los siguientes:

Tabla 4- Requisitos funcionales del sistema

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C1-Ob1	Búsqueda facetada por todas las propiedades	C1-BuscarArchivos	Crítico/Obligatorio	El sistema debe permitir una búsqueda facetada por todas y cada una de las propiedades de los archivos. Más adelante se definirán dichas propiedades generales y específicas de los archivos.
C1-Ob2	Modalidades en la búsqueda por palabras	C1-BuscarArchivos		<p>Para las búsquedas facetadas el sistema debe permitir una búsqueda por palabras en la que debe haber varias modalidades de búsqueda:</p> <ul style="list-style-type: none"> • buscar cualquier palabra de las introducidas • buscar todas las palabras introducidas en cualquier orden • buscar la frase exacta introducida <p>Este campo debe ser rellenado obligatoriamente.</p>

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C1-Ob3	Formas de ordenar los resultados	C1-BuscarArchivos	Crítico/Obligatorio	<p>Para las búsquedas facetadas el sistema debe ofrecer las siguientes opciones para ordenar los resultados:</p> <ul style="list-style-type: none"> • orden creciente de fecha en que se incluyó el archivo en el repositorio • orden decreciente • primero los archivos más descargados • por orden alfabético <p>Este campo debe ser rellenado obligatoriamente.</p>
C1-Ob4	Búsqueda no refinada por sistema operativo	C1-BuscarArchivos	Crítico/ Obligatorio	El Sistema debe mostrar una lista con todos los Sistemas Operativos incluidos y permitir que cualquier usuario pueda realizar una búsqueda no refinada de archivos que pertenezcan al Sistema Operativo elegido.
C1-Ob5	Búsqueda no refinada por tipo de archivo	C1-BuscarArchivos	Crítico/ Obligatorio	El Sistema debe mostrar una lista con todos los tipos de archivo incluidos y permitir que cualquier usuario pueda realizar una búsqueda no refinada de archivos que pertenezcan al tipo elegido.
C1-Ob6	Lista de los archivos más descargados	C1-BuscarArchivos	Crítico/ Obligatorio	El Sistema debe ofrecer una lista de archivos frecuentemente descargados.

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C1-Ob7	Lista de los archivos más recientes	C1-BuscarArchivos	Crítico/Obligatorio	El Sistema debe ofrecer una lista de archivos recientemente incluidos.
C1-Ob8	Búsqueda por inicial del archivo	C1-BuscarArchivos	Crítico/Obligatorio	El sistema debe ofrecer la opción de hacer una búsqueda por inicial.
C1-Op1	Búsqueda por tesauro	C1-BuscarArchivos	Crítico/Opcional	El Sistema podría incluir una búsqueda portesauro en la que se contemplen sinónimos, términos generales y términos específicos relativos al término buscado por el usuario.
C2-Ob1	Consulta de los detalles de un archivo	VerArchivos	Crítico/ Obligatorio	El Sistema debe permitir a cualquier usuario consultar la información asociada a un archivo.
C2-Ob2	Tipos de archivos permitidos	VerArchivos	Crítico/ Obligatorio	El Sistema debe incluir archivos de los siguientes tipos: DLL, OCX, DRV y FON.

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C2-Ob3	Campos generales a almacenar de un archivo	VerArchivos	Crítico/ Obligatorio	<p>El sistema debe almacenar la siguiente información general asociada a un archivo:</p> <ul style="list-style-type: none"> • <i>Nombre</i> • <i>Tipo de archivo</i> • <i>Tamaño</i> • <i>Fecha de subida del archivo al repositorio</i> • <i>Fecha de creación del archivo y antigüedad del archivo</i> • <i>Sistema Operativo bajo el que fue creado el archivo</i> • <i>Número de descargas</i> • <i>Datos de registro del Usuario que lo subió al repositorio excepto si es el administrador quien lo incluyó</i>

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C2-Ob4	Campos de versión y detalles de versión de un archivo	VerArchivos	Crítico/ Obligatorio	<p>El sistema debe almacenar la siguiente información específica de una DLL:</p> <ul style="list-style-type: none"> • Versión del archivo. • Versión del producto. • Subtipo de archivo. • Organización que ha creado el archivo. • Descripción de las funciones del archivo. • Nombre interno del archivo. • Copyright del archivo. • Marcas registradas del archivo. • Nombre original del archivo. • Nombre del producto en el que está incluido el archivo.
C2-Opl	Información de fiabilidad y errores de un archivo	VerArchivos	Crítico/Opcional	<p>El Sistema podría incluir información acerca de la fiabilidad del archivo, es decir los errores (bugs) que tiene, en base a un parseo de resultados obtenidos en google de comentarios en foros a esa DLL.</p>

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C2-Op2	Información de la frecuencia de uso de un archivo	VerArchivos	Crítico/ Opcional	El Sistema podría incluir información acerca de la frecuencia de uso de cada archivo en base un parseo de resultados obtenidos en google.
C2-Op3	Información de la seguridad de un archivo	VerArchivos	Crítico/ Opcional	El Repositorio podría incluir información acerca de la seguridad del archivo en base a un parseo de resultados obtenidos en google.
C3-Obl	Descarga de archivos	DescargarArchivo	Crítico/ Obligatorio	El Sistema debe permitir la descarga de cualquier archivo incluido a cualquier usuario.

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C4-Ob1	Registro de usuario	Registrarse	Crítico/Obligatorio	<p>El Sistema debe permitir que un usuario se registre con sus datos en el sistema. Para registrarse el usuario debe facilitar su <i>alias</i> de usuario, su <i>nombre</i>, sus <i>apellidos</i>, y su <i>email</i>. La contraseña la generara el sistema automáticamente y se la enviara al nuevo usuario por email. El usuario debe estar registrado antes de poder utilizar las siguientes funcionalidades:</p> <ul style="list-style-type: none"> • Subir comentario a archivo • Subir archivo • Enviar al foro una consulta o una respuesta a otra consulta. • Enviar sugerencia.
C4-Ob2	Campos obligatorios de registro	Registrarse	Crítico/Obligatorio	El sistema debe comprobar que ninguno de los datos queda en blanco.
C4-Ob3	Alias único	Registrarse	Crítico/Obligatorio	El sistema debe comprobar que el alias no lo posea otro usuario registrado anteriormente.

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C4-Ob4	Formato de los campos de registro.	Registrarse	Crítico/Obligatorio	<p>El sistema debe comprobar que el formato de los datos introducidos es correcto.</p> <ul style="list-style-type: none"> El alias no debe contener los siguientes caracteres: $\backslash [] ' ; : < > + = , ? *$ El nombre solo debe contener letras. El apellido solo debe contener letras. El email debe estar bien formado, es decir debe cumplir la siguiente expresión: $(\backslash w)^+ . (\backslash w)^+ * @ (\backslash w)^+ . (\backslash w)^+ * . (\backslash w)^+$
C5-Ob1	Modificar datos de registro	ModificarRegistro	Crítico/Obligatorio	El Sistema debe permitir que un usuario modifique sus datos de registro con las mismas restricciones que en el alta.
C5-Ob2	Alias no modificable	ModificarRegistro	Crítico/Obligatorio	El sistema no debe permitir modificar el alias.
C6-Ob1	Autenticación en funcionalidades restringidas	Autenticarse	Crítico/Obligatorio	El Sistema debe contemplar que un usuario se autentique con su alias y su contraseña antes de usar una de las funcionalidades restringidas.

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
C7-Ob1	Incluir nuevo archivo	SubirArchivo	Crítico/Obligatorio	El Sistema debe permitir que cualquier usuario previamente registrado pueda subir archivos y se incluyan en el repositorio.
C7-Ob2	Solo archivos de los tipos indicados	SubirArchivo	Crítico/Obligatorio	Solo se pueden enviar archivos de los tipos incluidos en el repositorio.
C8-Ob1	Eliminar archivo	EliminarArchivo	Crítico/Obligatorio	El Sistema debe permitir que un usuario administrador elimine un archivo que haya sido subido al repositorio.
C9-Ob1	Modificar archivo	ModificarArchivo	Crítico/Obligatorio	El Sistema debe permitir que un usuario administrador modifique las propiedades de un archivo que haya sido subido al repositorio.
C10-Ob1	Insertar comentario a archivo	SubirComentarioArchivo	Importante/Obligatorio	El Sistema debe permitir a cualquier usuario subir comentarios a cada archivo presente en el repositorio.
II-Ob1	Revisar comentarios a archivos	RevisarComentariosArchivo	Importante/Obligatorio	El Sistema debe ofrecer a un administrador la opción de obtener un listado para realizar la revisión de los comentarios asociados a archivos.
II-Ob2	Dar visto bueno a comentarios	RevisarComentariosArchivo	Importante/Obligatorio	El Sistema debe ofrecer a un administrador la opción de dar el visto a uno o varios comentarios.
II-Ob3	Eliminar comentarios a archivo	RevisarComentariosArchivo	Importante/Obligatorio	El Sistema debe ofrecer a un administrador la opción de eliminar uno o varios comentarios a archivo.

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
I2-Ob1	Foro de consultas	ConsultarForo	Importante/Obligatorio	El Sistema debe incluir un foro consistente en una lista de consultas.
I2-Ob2	Campos para una consulta	ConsultarForo	Importante/Obligatorio	<p>Por cada consulta se debe almacenar:</p> <ul style="list-style-type: none"> • <i>Título,</i> • <i>descripción ,</i> • <i>fecha en que se envió,</i> • <i>usuario que lo envió.</i>
I2-Ob3	Ordenar listado de consultas	ConsultarForo	Importante/Obligatorio	El listado se debe poder ordenar por <i>título</i> o por <i>fecha</i> .
I2-Ob4	Ver respuestas de una consulta	ConsultarForo	Importante/Obligatorio	Por cada consulta se almacenaran todas las respuestas. El sistema debe permitir que un usuario acceda a una consulta y vea todas las respuestas a dicha consulta

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
I2-Ob5	Campos de la respuesta	ConsultarForo	Importante/Obligatorio	<p>Por cada respuesta se almacenara:</p> <ul style="list-style-type: none"> • <i>Título</i>, • <i>Descripción</i>, • <i>Fecha</i> en que se envió, • <i>Usuario</i> que lo manda, • <i>Consulta</i> a la que responde.
I3-Ob1	Participación en el foro	ParticiparEnForo	Importante/Obligatorio	El Sistema debe ofrecer la posibilidad de participar en el foro a cualquier usuario ya sea para mandar una consulta como para contestar a una consulta existente.
I3-Ob1	Listado de los textos insertados en el foro	RevisarTextosForo	Importante/Obligatorio	El Sistema debe permitir que un administrador obtenga un listado de los textos subidos al foro tanto de consultas como respuestas para revisarlas.
I3-Ob2	Dar visto bueno a los textos del foro	RevisarTextosForo	Importante/Obligatorio	El Sistema debe permitir que un administrador dé el visto bueno a uno o varios textos del foro.
I3-Ob3	Eliminar textos del foro	RevisarTextosForo	Importante/Obligatorio	El Sistema debe permitir que un administrador elimine uno o varios textos del foro.

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
I4-Ob1	Sugerir un archivo	SugerirArchivo	Importante/Obligatorio	El Sistema debe permitir que cualquier usuario pueda pedir que se incluya un archivo que no haya encontrado, o sugerir cualquier otra cosa. El usuario debe indicar el tipo de sugerencia, si es una petición de un archivo o no, y el contenido de la sugerencia.
I5-Ob1	Listado de sugerencias	GestionarSugerencias	Importante/Obligatorio	El Sistema debe permitir que el administrador obtenga un listado de sugerencias de los usuarios. Para ello debe introducir o bien una <i>fecha desde</i> la cual el sistema devuelva la sugerencias recibidas o bien un <i>numero de sugerencias</i> a devolver.
I5-Ob2	Eliminar sugerencias	GestionarSugerencias	Importante/Obligatorio	El Sistema debe permitir que el administrador elimine una o varias sugerencias.
U1-Ob1	Listado de usuarios registrados	BloquearUsuarios	Útil/Obligatorio	El Sistema debe permitir que un administrador obtenga un listado de todos los usuarios registrados en el sistema.
U1-Ob2	Consulta de datos de registro por el administrador	BloquearUsuarios	Útil/Obligatorio	El Sistema debe permitir que un administrador consulte los datos de registro de un usuario. No debe permitir que el administrador los modifique.
U1-Ob3	Bloquear usuario	BloquearUsuarios	Útil/Obligatorio	El Sistema debe permitir que un administrador bloquee a uno o varios usuarios.
U2-Ob1	Añadir nuevo tipo de archivo al repositorio	AnadirTipoArchivo	Útil/Obligatorio	El Sistema debe permitir que un administrador añada un nuevo tipo de archivo al repositorio.

ID. Req.	Título	Caso de uso que lo incluye	Prioridad	Descripción del requisito
U3-Obl	Eliminar tipo de archivo	EliminarTipoArchivo	Útil/Obligatorio	El Sistema debe permitir que un administrador elimine un tipo de archivo y con el todos los archivos de ese tipo existentes en el repositorio
U4-Obl	Añadir nuevo sistema operativo	AnadirSO	Útil/Obligatorio	Debe permitir que un administrador añada un nuevo Sistema Operativo al repositorio.
U5-Obl	Eliminar sistema operativo	EliminarSO	Útil/Obligatorio	El Sistema debe permitir que un administrador elimine un Sistema Operativo y con el todos los archivos pertenecientes a Sistema.

5.1.6 Requisitos no funcionales

En esta sección se describen los requisitos no funcionales agrupados por calidades de sistema. Según lo hablado con el tutor del proyecto hemos considerado como importantes en este producto las calidades de *usabilidad*, *seguridad* y *rendimiento*.

Requisitos de usabilidad.

En primer lugar consideramos varias líneas generales en lo que se refiere a la usabilidad y accesibilidad del proyecto:

- Debe ser una aplicación accesible desde cualquier parte, es decir accesible para cualquier equipo con acceso Internet. Por tanto debe ser una aplicación Web.
- La interfaz de usuario debe ser consistente con las interfaces de los repositorios estudiados en el estado del arte, es decir que debe presentar una interfaz similar a estos y además recoger lo mejor de cada uno.
- Como ya hemos dicho la aplicación debe poder ser manejada por cualquier usuario que sepa manejar un ordenador, por lo tanto debe ser una interfaz muy intuitiva, donde haya facilidad de navegación.
- Como debe ser manejada por cualquier usuario la interfaz podría tener más de un lenguaje además del español, preferiblemente inglés, que es el idioma internacional. Este requisito se considera opcional y se tendría en cuenta en función del tiempo o para versiones futuras.
- El sistema debe mostrar los resultados paginados en las búsquedas, en los comentarios a los archivos, las consultas y respuestas del foro, etc. Se debe poder solicitar la página siguiente a la actual así como la anterior. Si un usuario está consultando la información asociada a un elemento el sistema debe mostrar la posibilidad de volver a la página actual de resultados desde donde se accedió al dicho elemento

Requisitos de seguridad

En esta aplicación como en la mayoría de las aplicaciones se requiere un control sobre quién tiene acceso a qué funcionalidades del sistema.

En este caso como ocurre usualmente los roles de seguridad iniciales vienen dados por los actores de los casos de uso. Por tanto se contemplan tres usuarios: *Usuario*, *UsuarioRegistrado* y *Administrador*.

Como puede verse en el diagrama de casos de uso, todos los casos de uso disponibles para un usuario estarán disponibles también para un usuario registrado. De la misma manera, los casos de uso de un usuario registrado estarán disponibles para el administrador. En consecuencia el administrador tendrá disponibles todos los casos de uso del sistema.

Casos de uso que no exigen seguridad

Para los casos de uso:

- *C1-BuscarArchivos*,
- *C2-VerArchivos*
- *C3 - DescargarArchivo*,
- *C4 - Registrarse*,
- *C6 – Autenticarse* y
- *I3 - ConsultarForo*.

no se describe ningún requisito de seguridad. Cualquiera que acceda a la aplicación debe poder disponer de todos ellos y no se le debe exigir estar registrado ni introducir datos de autenticación. Como hemos dicho estos casos de uso también deben estar disponibles para un usuario registrado y autenticado y para un administrador autenticado.

Casos de uso que requieren al menos un perfil de usuario registrado

Los casos de uso siguientes deben exigir autenticación antes de comenzar el caso de uso:

- C5-ModificarRegistro
- C7-SubirArchivo
- I1-SubirComentarioArchivo
- I4-ParticiparEnForo
- I5-SugerirArchivo

Deben estar visibles aunque el usuario no se haya autenticado todavía.

Casos de uso que requieren un perfil de administrador

Los casos de uso siguientes requieren perfil de administrador:

- C8-EliminarArchivo
- C9-ModificarArchivo
- I2-RevisarComentariosArchivo
- I5-RevisarTextosForo
- I7-GestionarSugerencias
- U1-BloquearUsuarios
- U2-AnadirTipoArchivo
- U3-EliminarTipoArchivo
- U4-AnadirSO
- U5-EliminarSO

Todos ellos no deben estar disponibles ni visibles en pantalla mientras el administrador no se autentique.

Casos de uso que requieren ser revisados por un administrador

Los requisitos de casos de uso que deberían ser revisados por un administrador son los siguientes:

- I1-SubirComentarioArchivo
- I4-ParticiparEnForo

Para la revisión del caso de uso *I1-SubirComentarioArchivo* se contempla el caso de uso *I2 - RevisarComentariosArchivo* y para revisar el caso de uso *I4-ParticiparEnForo* se contempla el caso de uso *I5 - RevisarTextosForo*.

Requisitos de rendimiento

El sistema debe ser escalable. Al aumentar el número de usuarios concurrentes el sistema mantendrá su tiempo de respuesta y rendimiento proporcionales al aumento de dichos usuarios.

Es decir que si el crecimiento del número de usuarios es lineal el crecimiento en el tiempo de respuesta o el decrecimiento en el rendimiento del sistema también deben ser lineales, tengamos el hardware que tengamos.

5.1.7 Glosario del Proyecto

Términos de dominio

Archivo: cada una de las unidades que componen el repositorio.

Administrador: persona que asume el rol de realizar tareas de mantenimiento del sistema

Registro: Datos de un usuario añadido al sistema.

Comentario: texto enviado por un usuario registrado en el que da una opinión acerca de un archivo.

Texto del foro: consulta del foro o respuesta a alguna consulta del foro enviada al sistema por un usuario registrado.

Sugerencia: mandada por un usuario, son de dos tipos:

- Sugerencia de archivo: petición de un archivo inexistente
- Otras sugerencias: resto

Sistema Operativo: Este dato se guarda para cada archivo con el fin de indicar cuando se puede utilizar.

Tipo de archivo: clase del archivo almacenado. En Windows viene dado necesariamente por la extensión del archivo. En Unix la extensión no es determinante.

Usuario: Cualquier persona que acceda a la aplicación desde cualquier parte.

Usuario registrado: Aquel que ha enviado sus datos y que, una vez autenticado puede utilizar las funcionalidades restringidas.

Sinónimos

Alias: en este proyecto se utiliza como sinónimo de login

Bajar: sinónimo de descargar del sistema

Criterio: sinónimo de palabra en la búsqueda facetada o refinada.

Password: termino en habla inglesa cuya traducción es contraseña.

Subir: sinónimo de enviar al sistema

Tema: Consulta (en el foro)

Términos de tecnología

Tesoro: La palabra **tesoro** se refiere al listado de palabras o términos empleados para representar conceptos. El término proviene del latín *thesaurus*, el cual tiene su origen del griego clásico *θησαυρός* (thesauros), *almacén*, *tesorería*.

En el campo de la bibliotecología, de las Ciencias de la Información y de las de la computación, un tesoro es una lista que contiene los “términos” empleados para representar los conceptos, temas o contenidos de los documentos, con miras a efectuar una normalización terminológica que permita mejorar el canal de acceso y comunicación entre los usuarios y las Unidades de Información.

Los términos que conforman el tesoro se interrelacionan entre ellos bajo tres modalidades de relación:

1. **Relaciones jerárquicas**: Establece subdivisiones que generalmente reflejan estructuras de Todo/Parte.

2. **Relaciones de equivalencia**: Controla la Sinonimia, Homonimia, Antonimia y Polinimia entre los términos.

3. **Relaciones asociativas:** Mejoran las estrategias de recuperación y ayudan a reducir la polijerarquía entre los términos [11].

Búsqueda facetada: búsqueda de elementos en base a cualquiera de sus campos o propiedades almacenadas.

Términos de desarrollo de software

Bugs: Errores en un programa, el programa deja de hacer lo que se le supone que haga.

Calidades del producto: determinan lo bien que se comporta el sistema en el límite entre el actor y el sistema. Se agrupan principalmente en tres categorías: **como se manifiesta** el sistema, **como funciona** y **cómo evoluciona** en el tiempo. Las calidades más importantes son:

- **Cómo se manifiesta:** comportamiento, disponibilidad y usabilidad.
- **Cómo funciona:** rendimiento, manejabilidad, seguridad, robustez y fiabilidad.
- **Cómo evoluciona :** escalabilidad, mantenibilidad, extensibilidad, flexibilidad, reusabilidad y portabilidad [12].

Casilla de verificación: en una interfaz de usuario una casilla de verificación es un control en el que se puede hacer clic para marcar o desmarcar una sección con lo que indicamos que queremos que se tenga en cuenta o no la funcionalidad expresada en dicha sección.

Home: Pantalla principal de una aplicación Web.

Login: termino asociado a un usuario, escogido por él, con el que se le identifica en el sistema.

Parseo: interpretación automática de un texto.

Plataforma: puede ser de dos tipos:

- **Plataforma hardware:** arquitectura hardware subyacente de un equipo informático.

- Plataforma software: sistema operativo subyacente instalado en un equipo informático.
- **Rendimiento**: medida que indica la cantidad de trabajo por unidad de tiempo de un sistema. Se puede medir en muestras por segundo o en Kilobytes por segundo. En nuestro caso una muestra sería una petición de página web.
- **Tiempo de respuesta**: medida que indica la cantidad de tiempo que tarda un sistema en cumplir la orden que se le pide. En nuestro caso es el tiempo que tarda en entregarnos una página web.

5.2 Flujo de Análisis

A continuación se expone el análisis de cada caso de uso y el conjunto de objetos resultante que forman parte de nuestro modelo de negocio

5.2.1 Análisis de los casos de uso

Tabla 5- Análisis caso de uso C1-BuscarArchivos

ID y nombre	<i>C1:BuscarArchivos</i>
Descripción	Un usuario realiza una búsqueda por criterios o bien una búsqueda simple por Sistema Operativo determinado o bien de un determinado tipo.
Actores Principales	Usuario o UsuarioRegistrado o Administrador.
Actores Secundarios	Ninguno.
Prioridad	Crítico.
Precondiciones	Ninguna.
Disparador	El caso de uso empieza cuando el usuario escoge una de las formas de búsqueda de archivos.
Flujo principal	<ol style="list-style-type: none"> 1. Si el usuario escoge realizar una búsqueda facetada <ol style="list-style-type: none"> 1.1. el usuario pulsa el botón de búsqueda avanzada 1.2. mientras los datos no son correctos <ol style="list-style-type: none"> 1.2.1. el sistema muestra el formulario de búsqueda avanzada 1.2.2. el usuario rellena los datos 1.2.3. el usuario pulsa aceptar 1.2.4. sistema comprueba los datos introducidos 1.2.5. Si parámetros incorrectos <ol style="list-style-type: none"> 1.2.5.1. sistema muestra errores 2. Si no, si el usuario escoge pedir un listado de archivos por Sistema Operativo

	<p>2.1. El usuario pulsa el enlace del sistema operativo deseado.</p> <p>3. Si no, Si el usuario escoge pedir un listado de archivos por tipo de archivo</p> <p>3.1. El usuario pulsa el enlace del tipo operativo deseado.</p> <p>4. Sistema realiza la búsqueda.</p> <p>5. Si existen archivos que cumplan lo requerido:</p> <p>5.1. El sistema devuelve los archivos correspondientes a ese tipo de forma paginada.</p> <p>6. Si no</p> <p>6.1. El sistema muestra el mensaje correspondiente.</p>
Flujos alternativos	Ninguno.
Postcondiciones	Ninguna.
Requisitos no funcionales	

Tabla 6- Análisis del caso de uso C2-VerArchivos

ID y nombre	<i>C2:VerArchivos</i>
Descripción	Un usuario navega por los resultados y consulta los detalles de algún archivo. Para ello realiza una búsqueda ejecutando el caso de uso <i>C1-BuscarArchivos</i> .
Actores Principales	Usuario o UsuarioRegistrado o Administrador.
Actores Secundarios	Ninguno.
Prioridad	Crítico.
Precondiciones	Si el usuario navega por los resultados primero debe ejecutar el caso de uso <i>C1-BuscarArchivos</i> y la búsqueda debe devolver resultados.
Disparador:	El caso de uso empieza cuando el usuario pulsa en un enlace de un archivo reciente o más descargado o bien tras realizar una búsqueda y solicitar una página o ver los detalles de un archivo de los resultados.
Flujo principal:	<ol style="list-style-type: none"> 1. Si el usuario decide ver archivos recientes o más descargados: <ol style="list-style-type: none"> 1.1. el usuario pulsa en el enlace del archivo correspondiente 1.2. el sistema muestra los detalles del archivo 2. Si el usuario hace una búsqueda y pulsa en un enlace de los resultados <ol style="list-style-type: none"> 2.1. el sistema muestra la página de información del archivo.
Postcondiciones	Ninguna.
Requisitos no funcionales	

Tabla 7- Análisis del caso de uso C3-DescargarArchivo

ID y nombre	<i>C3:DescargarArchivo</i>
Descripción	Un usuario solicita la descarga de un archivo.
Actores Principales	Usuario o UsuarioRegistrado o Administrador.
Actores Secundarios	Ninguno.
Prioridad	Crítico.
Precondiciones	El usuario debe navegar a la vista de un archivo.
Disparador:	El caso de uso empieza tras ejecutar un búsqueda y seleccionar un archivo presionando el botón Descargar.
Flujo principal:	<ol style="list-style-type: none"> 1. El usuario realiza una búsqueda y selecciona un archivo (casos C1 y C2). 2. El usuario pulsa el botón Descargar. 3. El sistema presenta el cuadro de diálogo para descargar archivo. 4. Si el usuario pulsa Aceptar: <ol style="list-style-type: none"> 4.1. empieza la descarga. 4.2. Si la operación se completa con éxito <ol style="list-style-type: none"> 4.2.1. el sistema informa de que se ha completado la operación con éxito. 4.3. Si no <ol style="list-style-type: none"> 4.3.1. el sistema informa del error.
Postcondiciones	Ninguna.
Requisitos no funcionales	

Tabla 8- Análisis del caso de uso C4-Registrarse

ID y nombre	<i>C4:Registrarse</i>
Descripción	El usuario rellena sus datos de registro y los envía para que el sistema incluya el nuevo usuario.
Actores Principales	Usuario.
Actores Secundarios	Ninguno.
Prioridad	Crítico
Precondiciones	Ninguna.
Disparador:	El caso de uso empieza cuando el usuario escoge la funcionalidad de registrarse ya sea desde el home como desde la pantalla de autenticación.
Flujo principal:	<ol style="list-style-type: none"> 1. Usuario pulsa el botón registrarse. 2. Mientras los datos no son correctos y el usuario no cancela <ol style="list-style-type: none"> 2.1. El sistema presenta un formulario para que el usuario introduzca su alias de usuario, el nombre, apellidos e email. 2.2. El usuario introduce todos los datos. 2.3. Si acepta el envío. <ol style="list-style-type: none"> 2.3.1. El sistema comprueba el alias 2.3.2. Si está mal formado <ol style="list-style-type: none"> 2.3.2.1. El sistema muestra el error 2.3.3. El sistema comprueba el nombre 2.3.4. Si está en blanco o no contiene solo letras <ol style="list-style-type: none"> 2.3.4.1. El sistema almacena el error 2.3.5. El sistema comprueba el campo apellidos 2.3.6. Si está en blanco o no solo contiene letras <ol style="list-style-type: none"> 2.3.6.1. el sistema almacena el error 2.3.7. El sistema comprueba el email

	<p>2.3.8. Si está en blanco o no está bien formado</p> <p>2.3.8.1. El sistema almacena el error</p> <p>2.3.9. Si hay algún error</p> <p>2.3.9.1. El sistema informa de los errores que haya</p> <p>2.3.10. Si no</p> <p>2.3.10.1. El sistema comprueba si existe el alias</p> <p>2.3.10.2. Si existe</p> <p>2.3.10.2.1. El sistema muestra el mensaje de error</p> <p>2.3.10.3. Si no existe</p> <p>2.3.10.3.1. El sistema genera una contraseña y se la manda al usuario por email.</p> <p>2.3.10.3.2. El sistema almacena al nuevo usuario.</p> <p>2.3.10.3.3. Si se almacena correctamente</p> <p>2.3.10.3.3.1. El sistema informa del éxito en la operación.</p> <p>2.3.10.3.4. Si no</p> <p>2.3.10.3.4.1. El sistema informa del error</p>
Flujos alternativos	Ninguno.
Postcondiciones	El sistema ha almacenado correctamente los datos del nuevo usuario. El usuario podrá autenticarse en cuanto le llegue la contraseña a la cuenta de correo facilitada.
Requisitos no funcionales	

Tabla 9- Análisis del caso de uso C5-Modificar Registro

ID y nombre	C5:ModificarRegistro
Descripción	El usuario modifica sus datos de registro.
Actores Principales	Usuario.
Actores Secundarios	Ninguno.
Prioridad	Crítico.
Precondiciones	Ninguna
Disparador:	El caso de uso comienza cuando el usuario decide utilizar la funcionalidad de <i>Modificar Registro</i> pulsando el botón correspondiente en el home.
Flujo principal:	<ol style="list-style-type: none"> 1. El usuario pulsa el botón <i>Modificar Registro</i> 2. El sistema presenta un formulario para modificar los datos de registro 3. Mientras datos no correctos y usuario no cancela: <ol style="list-style-type: none"> 3.1. usuario rellena los datos del formulario. 3.2. usuario pulsa enviar 3.3. sistema comprueba autenticación 3.4. si usuario no autenticado <ol style="list-style-type: none"> 3.4.1. sistema ejecuta el caso de uso C6-Autenticarse 3.5. sistema comprueba si usuario ya se ha autenticado <ol style="list-style-type: none"> 3.5.1. si usuario autenticado <ol style="list-style-type: none"> 3.5.1.1. El sistema muestra un formulario para modificar los datos. 3.5.1.2. El usuario introduce los nuevos datos de Registro <ol style="list-style-type: none"> 3.5.1.2.1. Si el usuario quiere <ol style="list-style-type: none"> 3.5.1.2.1.1. El usuario introduce una nueva contraseña. 3.5.1.2.2. Si el usuario quiere <ol style="list-style-type: none"> 3.5.1.2.2.1. El usuario introduce el email. 3.5.1.3. Si el usuario acepta el envío. <ol style="list-style-type: none"> 3.5.1.3.1. El sistema comprueba la contraseña

	<p>3.5.1.3.2. Si está en blanco o tiene menos de 8 caracteres</p> <p>3.5.1.3.2.1. El sistema almacena el error</p> <p>3.5.1.3.3. El sistema comprueba el email</p> <p>3.5.1.3.4. Si está en blanco o mal formado</p> <p>3.5.1.3.4.1. El sistema almacena el error</p> <p>3.5.1.3.5. Si hay algún error</p> <p>3.5.1.3.5.1. El sistema informa de los errores que haya</p> <p>3.5.1.3.6. Si no</p> <p>3.5.1.3.6.1. el sistema guarda los cambios.</p>
Flujos alternativos	Ninguno.
Postcondiciones	Ninguna.
Requisitos no funcionales	

Tabla 10- Análisis del caso de uso C6-Autenticarse

ID y nombre	<i>C6:Autenticarse</i>
Descripción	Una persona introduce su usuario y su contraseña para poder utilizar las funcionalidades de usuario registrado o de administrador.
Actores Principales	Usuario.
Actores Secundarios	Ninguno.
Prioridad	Crítico.
Precondiciones	Ninguna.
Disparador	El caso de uso comienza cuando un usuario escoge la funcionalidad de autenticarse o bien cuando escoge una funcionalidad restringida y aun no se ha autenticado.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de Autenticarse. 2. Mientras los datos no son correctos y usuario no pulsa Registrarse y usuario no cancela <ol style="list-style-type: none"> 2.1. El sistema presenta un formulario para introducir usuario y contraseña de registro. En el formulario también se incluirá un enlace para registrarse. 2.2. Si el usuario está registrado <ol style="list-style-type: none"> 2.2.1. El usuario introduce los datos 2.2.2. El usuario acepta el envío 2.2.3. El sistema comprueba los datos. 2.2.4. Si son correctos <ol style="list-style-type: none"> 2.2.4.1. El usuario es autenticado 2.2.5. Si no <ol style="list-style-type: none"> 2.2.5.1. El sistema presenta el mensaje de error 2.3. Si no <ol style="list-style-type: none"> 2.3.1. El usuario solicita el formulario de registro pulsando el botón registrarse en el formulario de autenticación. 2.3.2. Se ejecuta el caso de uso registrarse.

Flujos alternativos	Ninguno.
Postcondiciones	El usuario esta autenticado y podrá utilizar las funciones de usuario registrado o de administrador según el perfil introducido.
Requisitos no funcionales	

Tabla 11- Análisis del caso de uso C7-SubirArchivo

ID y nombre	<i>C7:SubirArchivo</i>
Descripción	El usuario sube un archivo desde su equipo al repositorio.
Actores Principales	Usuario registrado
Actores Secundarios	Ninguno.
Prioridad	Crítico
Precondiciones	<ol style="list-style-type: none"> 1. El archivo enviado tiene que ser de uno de los tipos contemplados. 2. El usuario debe estar registrado y autenticado.
Disparador	El caso de uso empieza cuando el usuario escoge la opción de subir archivo en el home pulsando el botón correspondiente.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón Subir Archivo 2. si el usuario no se ha autenticado <ol style="list-style-type: none"> 2.1. el sistema pide que se autentique ejecutando <i>C6-Autenticarse</i>. 3. si el usuario ya se ha autenticado: <ol style="list-style-type: none"> 3.1. el sistema presenta un formulario para introducir el archivo <ol style="list-style-type: none"> 3.1.1. el usuario pulsa Seleccionar Archivo. 3.1.2. el usuario elige un archivo del disco duro. 3.1.3. el usuario pulsa enviar 3.2. si el usuario acepta el envío. <ol style="list-style-type: none"> 3.2.1. el sistema comprueba el archivo 3.2.2. si está en blanco o no es uno de los permitidos <ol style="list-style-type: none"> 3.2.2.1. El sistema muestra error 3.2.3. Si no <ol style="list-style-type: none"> 3.2.3.1. El usuario envía el fichero 3.2.3.2. Si llega correctamente <ol style="list-style-type: none"> 3.2.3.2.1. El sistema almacena el archivo. 3.2.3.2.2. El sistema indiza la información del archivo. 3.2.3.2.3. El sistema informa del éxito.

	3.2.3.3. Si no 3.2.3.3.1. El sistema informa del error.
Flujos alternativos	
Postcondiciones	El archivo se almacena correctamente en el sistema.
Requisitos no funcionales	

Tabla 12- Análisis del caso de uso C8-EliminarArchivo

ID y nombre	<i>C8:EliminarArchivo</i>
Descripción	Un usuario administrador elimina un archivo del sistema.
Actores Principales	Usuario Administrador.
Actores Secundarios	Ninguno.
Prioridad	Crítico
Precondiciones	El usuario debe estar autenticado.
Disparador	El caso de uso empieza cuando el usuario escoge una de las formas de buscar archivos.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador accede a la vista de un archivo. 2. El administrador pulsa el botón Eliminar Archivo. <ol style="list-style-type: none"> 2.1. el sistema pide confirmación. <ol style="list-style-type: none"> 2.1.1. si el usuario pulsa aceptar <ol style="list-style-type: none"> 2.1.1.1. el sistema procede a eliminar el archivo. 2.1.2. si la operación se completa con éxito <ol style="list-style-type: none"> 2.1.2.1. el sistema informa del éxito 2.1.3. si no <ol style="list-style-type: none"> 2.1.3.1. el sistema informa del error
Flujos alternativos	Ninguno.
Postcondiciones	El sistema debe haber eliminado el archivo.
Requisitos no funcionales	

Tabla 13- Análisis del caso de uso C9-ModificarArchivo

ID y nombre	C9:ModificarArchivo
Descripción	Un usuario administrador modifica las propiedades de un archivo del sistema.
Actores Principales	Usuario Administrador.
Actores Secundarios	Ninguno.
Prioridad	Crítico
Precondiciones	El usuario debe estar autenticado.
Disparador	El caso de uso empieza cuando el usuario escoge una de las formas de buscar archivos.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador accede a la vista de un archivo. 2. El administrador pulsa el botón Modificar Archivo. 3. El sistema muestra un formulario con las propiedades del archivo. 4. El usuario realiza las modificaciones oportunas y pulsa aceptar <ol style="list-style-type: none"> 4.1. el sistema procede a modificar el archivo. 4.2. si la operación se completa con éxito <ol style="list-style-type: none"> 4.2.1. el sistema informa del éxito 4.3. si no <ol style="list-style-type: none"> 4.3.1. el sistema informa del error
Flujos alternativos	Ninguno.
Postcondiciones	El sistema debe haber eliminado el archivo.
Requisitos no funcionales	

Tabla 14- Análisis del caso de uso C10-SubirComentarioArchivo

ID y nombre	<i>C10:SubirComentarioArchivo</i>
Descripción	Un usuario registrado añade un comentario a un archivo del repositorio.
Actores Principales	Usuario registrado
Actores Secundarios	Ninguno.
Prioridad	Crítico
Precondiciones	El usuario debe estar autenticado.
Disparador	El caso de uso empieza cuando el usuario escoge una de las formas de buscar archivos.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede a la vista de un archivo. 2. El usuario pulsa el botón Insertar Comentario. 3. Si el usuario no se ha autenticado <ol style="list-style-type: none"> 3.1.1. el sistema pide que se autentique ejecutando el caso de uso C6- <i>Autenticarse</i>. 4. Si el usuario esta autenticado <ol style="list-style-type: none"> 4.1. el sistema muestra un formulario con las propiedades del archivo. 4.2. El usuario rellena los datos y pulsa enviar 4.3. Si faltan el datos <ol style="list-style-type: none"> 4.3.1. El sistema vuelve a mostrar el formulario 4.4. Si están correctos <ol style="list-style-type: none"> 4.4.1. El sistema almacena los datos. 4.4.2. si la operación se completa con éxito <ol style="list-style-type: none"> 4.4.2.1. el sistema informa del éxito 4.4.3. si no <ol style="list-style-type: none"> 4.4.3.1. el sistema informa del error

Flujos alternativos	Ninguno.
Postcondiciones	El sistema debe haber eliminado el archivo.
Requisitos no funcionales	

5.2.2 Modelo de negocio

Tras el análisis de los casos de uso en los cuales se define lo que el sistema hace para los usuarios, ya estamos en posición de determinar cuáles van a ser los objetos más importantes de nuestro sistema que van a realizar dicho trabajo y van a almacenar los datos que el sistema necesita, es decir el modelo de negocio. Se presenta a continuación un diagrama de clases de análisis con dicho modelo de negocio, en el cual se pueden apreciar las clases fundamentales del sistema y las relaciones existentes entre ellas.

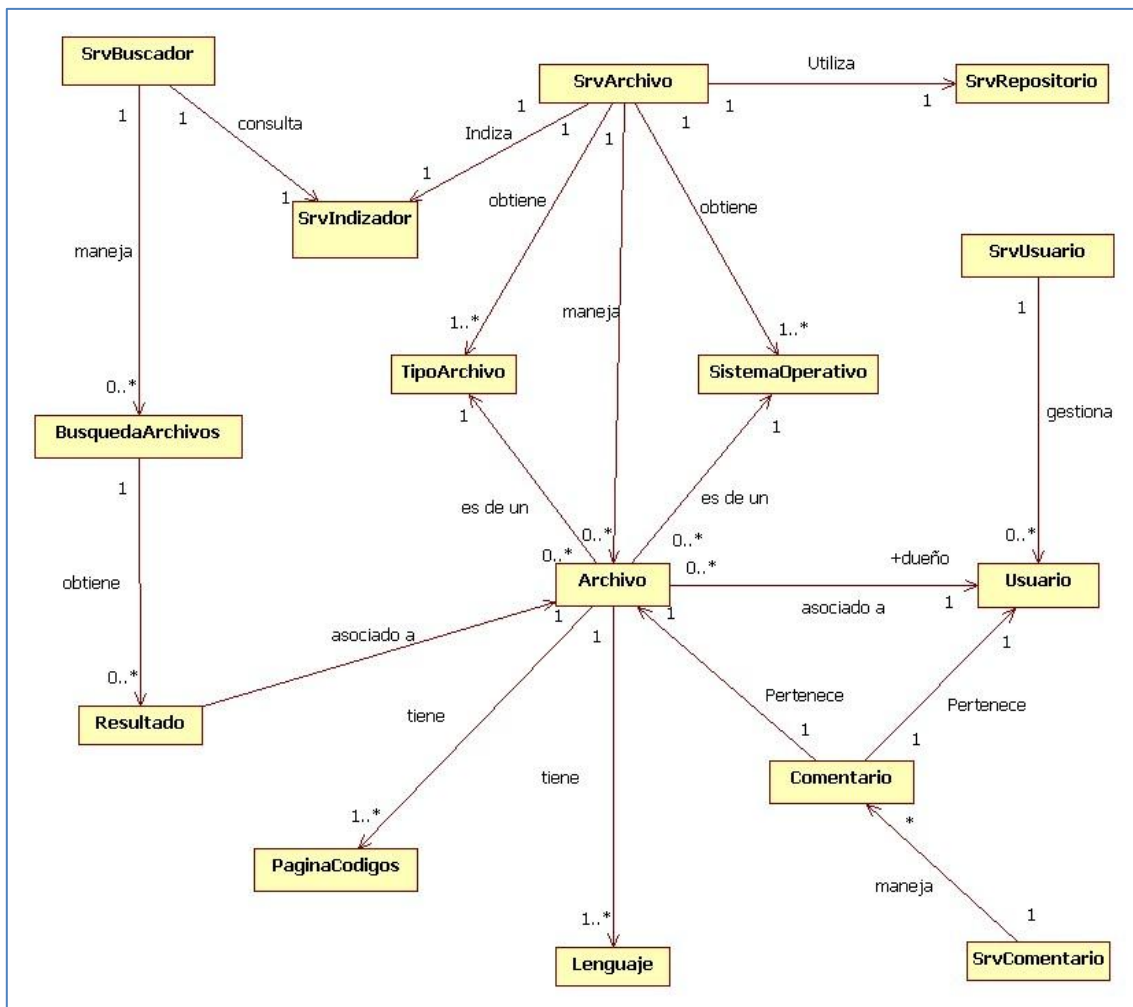


Ilustración 16- Diagrama de clases de análisis

SrvArchivo

Clase que contiene las operaciones relacionadas con los archivos. Esta clase sólo va a tener 1 instancia. Estos son los métodos principales:

Tabla 15- Definición de la clase SrvArchivo

SrvArchivo	
Métodos	
nuevoArchivo	Alta de un archivo
modificarArchivo	Modificación de un archivo
borrarArchivo	Eliminación de un archivo
buscarArchivoPor Id	Devuelve un objeto archivo a partir de su id
buscarSistemaOperativo	Devuelve un objeto SistemaOperativo a partir de su id
buscarTipoArchivo	Devuelve un objeto TipoArchivo a partir de su id
buscarLenguaje	Devuelve un objeto Lenguaje a partir de su id
buscarPaginaCodigos	Devuelve un objeto de tipo PaginaCodigos a partir de su id
devMasRecientes	Devuelve una lista de archivos conteniendo los más recientemente incluidos en el sistema
devMasDescargados	Devuelve una lista de archivos conteniendo los más descargados del sistema
devListaSSOO	Devuelve una lista con los Sistemas Operativos existentes en el sistema
devListaTipos	Devuelve una lista con los tipos de archivo permitidos en el sistema
devListaLenguajes	Devuelve una lista con los lenguajes existentes
devListaPaginaCodigos	Devuelve una lista con las páginas de códigos existentes

SrvRepositorio

Clase que contiene las operaciones relacionadas con el repositorio físico. Esta clase solo va a tener 1 instancia. Métodos principales:

Tabla 16- Definición de la clase SrvRepositorio

SrvRepositorio	
Métodos	
asignarRuta	Devuelve una ruta en el repositorio físico para almacenar un nuevo archivo
escribirFicheroAdisco	Copia el fichero desde memoria principal al disco duro
borrarFicheroFisico	Elimina un fichero físico del repositorio

SrvBuscador

Clase que proporciona los resultados relacionados con la búsqueda realizada. Esta clase solo va a tener 1 instancia.

Tabla 17- Definición de la clase SrvBuscador

SrvBuscador	
Métodos	
buscarArchivos	Devuelve los archivos que coincidan con los parámetros de búsqueda, sea esta del tipo que sea. Este método soporta todos los tipos de búsqueda de la aplicación: por palabras, por tipo, por sistema operativo y avanzada
devLenguajesSoportados	Devuelve una lista de lenguajes de búsqueda posibles para rellenar los parámetros
devOrdenesSoportados	Devuelve una lista de posibles formas de ordenar los resultados

SrvIndizador

Clase que proporciona la lectura y escritura del índice de archivos.

Tabla 18- Definición de la clase SrvIndizador

SrvBuscador	
Métodos	
indizarArchivo	Clasifica la información de un archivo introduciéndola en el índice
obtenerDocumentos	Devuelve una lista de documentos del índice que coinciden con los criterios de búsqueda.

SrvUsuario

Clase que proporciona las operaciones relacionadas con los usuarios y la autenticación. Esta clase solo va a tener 1 instanciación.

Tabla 19- Contenido de la clase SrvUsuario

SrvUsuario	
Métodos	
autenticar	Método que nos dice si un usuario y contraseña son validos
nuevoUsuario	Alta de usuario
modificarUsuario	Modificación de datos de usuario
buscarPorLogin	Devuelve un objeto Usuario por id
existeUsuario	Método que indica si un nombre de usuario ya está siendo usado

SrvComentario

Clase que proporciona las operaciones relacionadas con los comentarios a archivos. Esta clase solo va a tener una instanciación.

Tabla 20- Contenido de la clase SrvComentario

SrvComentario	
Métodos	
insertarComentario	Método para dar de alta un comentario
obtenerComentarios	Método para obtener los comentarios de un archivo

Archivo

Clase que contiene toda la información de los archivos (facetas). Aquí se muestran únicamente las propiedades generales pero en la implementación se debe incluir también toda la información de versión y detalles de la versión, contenida dentro del archivo físico.

Tabla 21- Contenido de la clase Archivo

Archivo	
Atributos	
id	Identificador, clave primaria
nombre	Nombre del archivo sin la extensión
pathUrl	Ruta donde del repositorio físico donde está el archivo almacenado
numDescargas	Contador con el número de veces que se ha descargado el archivo
tamaño	Tamaño en bytes
fechasubida	Fecha en que fue incluido en el sistema
propietario	Usuario que lo incluyó en el sistema
extensión	Extensión del nombre del archivo
versiónArchivo	Versión del archivo
versiónProducto.	Versión del producto al que pertenece el archivo

sistemaOperativo	Sistema Operativo para el que se diseñó el archivo
tipoArchivo	Tipo del archivo
subtipoArchivo	Si el tipo es Driver o Font se almacena el subtipo
fechaCreacion	Fecha de creación del archivo
organización	Compañía que creo el archivo
descripción	Descripción de la funcionalidad del archivo
nombreInterno	Nombre interno del archivo
copyright	Copyright del archivo
marcasRegistradas	Marcas registradas del archivo
nombreOriginal	Nombre original del archivo
nombreProducto	Nombre del producto al que pertenece el archivo
valoracionMedia	Media de valoraciones de los usuarios
dificultadUsoMedia	Media de valoraciones de los usuarios de dificultad de uso del archivo
seguridadMedia	Media de valoraciones de los usuarios de la seguridad del archivo
documentacionMedia	Media de valoraciones de los usuarios de la documentación existente acerca del uso del archivo
fiabilidadMedia	Media de valoraciones de los usuarios de la fiabilidad del archivo
comentarioGeneral	Comentario del administrador del sistema

TipoArchivo

Clase para almacenar los tipos de archivo permitidos.

Tabla 22- Contenido de la clase TipoArchivo

TipoArchivo	
Atributos	
id	Identificador, clave primaria
descripción	Descripción del tipo de archivo

SistemaOperativo

Clase para almacenar los sistemas operativos a los pertenecen los archivos.

Tabla 23- Contenido de la clase SistemaOperativo

Sistema Operativo	
Atributos	
id	Identificador, clave primaria
descripción	Descripción del sistema operativo

BusquedaArchivos

Clase que almacena los parámetros de búsqueda. Se especifican las propiedades generales de búsqueda, pero además a la hora de la implementación deberá incluir todas las facetas incluidas en el archivo (datos generales, datos de versión y detalles de versión) por las que el sistema debe clasificar un archivo.

Tabla 24- Contenido de la clase BusquedaArchivos

BusquedaArchivos	
Atributos	
tipoBusqueda	Búsqueda por palabras, por sistema operativo, por tipo o avanzada
resultados	Colección de archivos que cumplen con los parámetros de búsqueda
algunaPalabra	Campo para buscar resultados que contengan alguna de las palabras incluidas en el.
todasPalabras	Campo para buscar resultados que contengan todas las palabras incluidas en el.
fraseExacta	Campo para buscar resultados que contengan la frase exacta incluida en el.
sinLasPalabras	Campo para buscar resultados que no contengan ninguna de las palabras incluidas en el.
lenguajeBusqueda	Campo para indicar en que lenguaje están las palabras introducidas
orden	Campo para indicar en qué orden se desea recuperar los resultados

Resultado

Cada objeto de esta clase representa un archivo obtenido en una búsqueda, de ahí la relación 1 a muchos entre BusquedaArchivos y Resultado. Esta clase solo contiene el id del archivo al que apunta.

PaginaCodigos

Clase para almacenar los diferentes tipos de Páginas de Códigos con las que están codificados los detalles de los archivos.

Tabla 25- Contenido de la clase PaginaCodigos

PaginaCodigos	
Atributos	
id	Identificador, clave primaria
descripción	Descripción de la página de códigos

Lenguaje

Clase para almacenar los diferentes lenguajes con los que pueden estar escritos los detalles de los archivos.

Tabla 26- Contenido de la clase Lenguaje

Lenguaje	
Atributos	
id	Identificador, clave primaria
lenguaje	Descripción del lenguaje

Comentario

Texto y valoración de un usuario a un archivo, de ahí su relación con 1 usuario y 1 archivo.

Tabla 27- Contenido de la clase Comentario

Comentario	
Atributos	
id	Identificador del comentario
archivo	Archivo que se valora
usuario	Usuario que ha enviado la valoración
título	Resumen de la valoración
texto	Contenido de la opinión del usuario
valoración	Valoración general del archivo
dificultadUso	Valoración del usuario de la dificultad de uso del archivo
seguridad	Valoración del usuario de la seguridad del archivo
documentación	Valoración del usuario de la documentación de uso del archivo
urlDocumentacion	Url donde se puede encontrar documentación sobre el archivo

Usuario

Clase para almacenar los datos de un usuario que accede al sistema.

Tabla 28- Contenido de la clase Usuario

Usuario	
Atributos	
usuario	Identificador del usuario
nombre	Nombre del usuario
apellidos	Apellidos del usuario
email	Email del usuario
perfil	Usuario registrado o Administrador
password	Contraseña del usuario

5.3 Flujo de Diseño

El modelo de diseño implementa los requerimientos funcionales del sistema. Cada modelo de diseño es una realización de un caso de uso y describe los tipos de componentes requeridos y sus relaciones para representar el caso de uso.

El diseño nos ayuda a modelar como el sistema va a soportar los procesos de negocio. El modelo de diseño consta de tres tipos de componentes:

- Boundary (frontera) – Interfaz de usuario. Componentes que interaccionan con el usuario. Acceden a funciones de componentes de servicio y entidad.
- Service (servicio) – componentes que realizan las operaciones de negocio y el control del flujo del caso de uso. En este caso el estereotipo UML es <<control>>, pero no es una clase controladora.
- Entity (entidad) - componentes que se corresponden con las entidades del modelo de dominio[13].

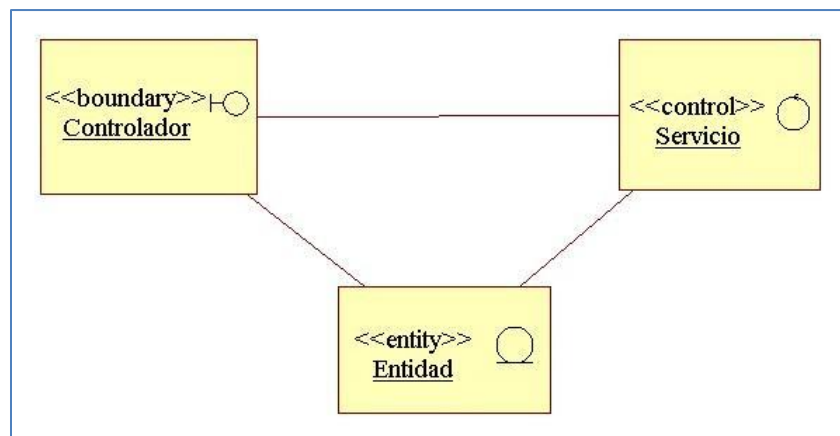


Ilustración 17- Tipos de componentes de diseño

La forma escogida de realizar el modelo de diseño será mediante diagramas de secuencia de UML. En algunos casos serán necesarios varios diagramas para explicar un caso de uso, los más complejos, y en otros casos basta con un solo diagrama.

5.3.1 Modelo de diseño para C1-BuscarArchivos

El caso de uso *C1-BuscarArchivos* se describe en tres diagramas de secuencia. En el primer diagrama se plasma la búsqueda facetada. En el segundo diagrama se trata la búsqueda no refinada, es decir por Sistema Operativo o por Tipo. Como ejemplo se ilustra solo la búsqueda por *Sistema Operativo*, ya que son similares. En el tercer diagrama se muestra la navegación por los resultados, parte común a ambos tipos de búsquedas. Como ejemplo se muestra navegar a la página siguiente.

En este caso de uso, para poder mostrar los resultados de búsqueda paginados se ha decidido crear un registro en memoria principal de búsquedas y resultados de esas búsquedas. Se almacena solo una búsqueda por sesión de usuario así como el estado de cada búsqueda y los resultados correspondientes.

Cada sesión de usuario caducará tras un tiempo de inactividad. Al cumplirse, la sesión expirará y con ella se eliminará la última búsqueda que haya hecho el usuario. Si el usuario en la misma sesión hace más de una búsqueda se sobrescribirá la anterior. Solo se guarda la última búsqueda.

Búsqueda es una entidad no persistente, es decir que no se contempla almacenar las búsquedas en la base de datos.

Dentro de cada búsqueda se almacenarán los resultados de la misma, mediante punteros a los archivos del repositorio. Con ello conseguimos no guardar el mismo archivo varias veces, si está presente en varias búsquedas.

Para la búsqueda facetada se ha pensado que el usuario solicite al sistema un formulario en una ventana aparte y en él rellenará todos los datos. Si hubiera errores se volvería a mostrar el formulario incluyendo en él los mensajes correspondientes a todos los errores.

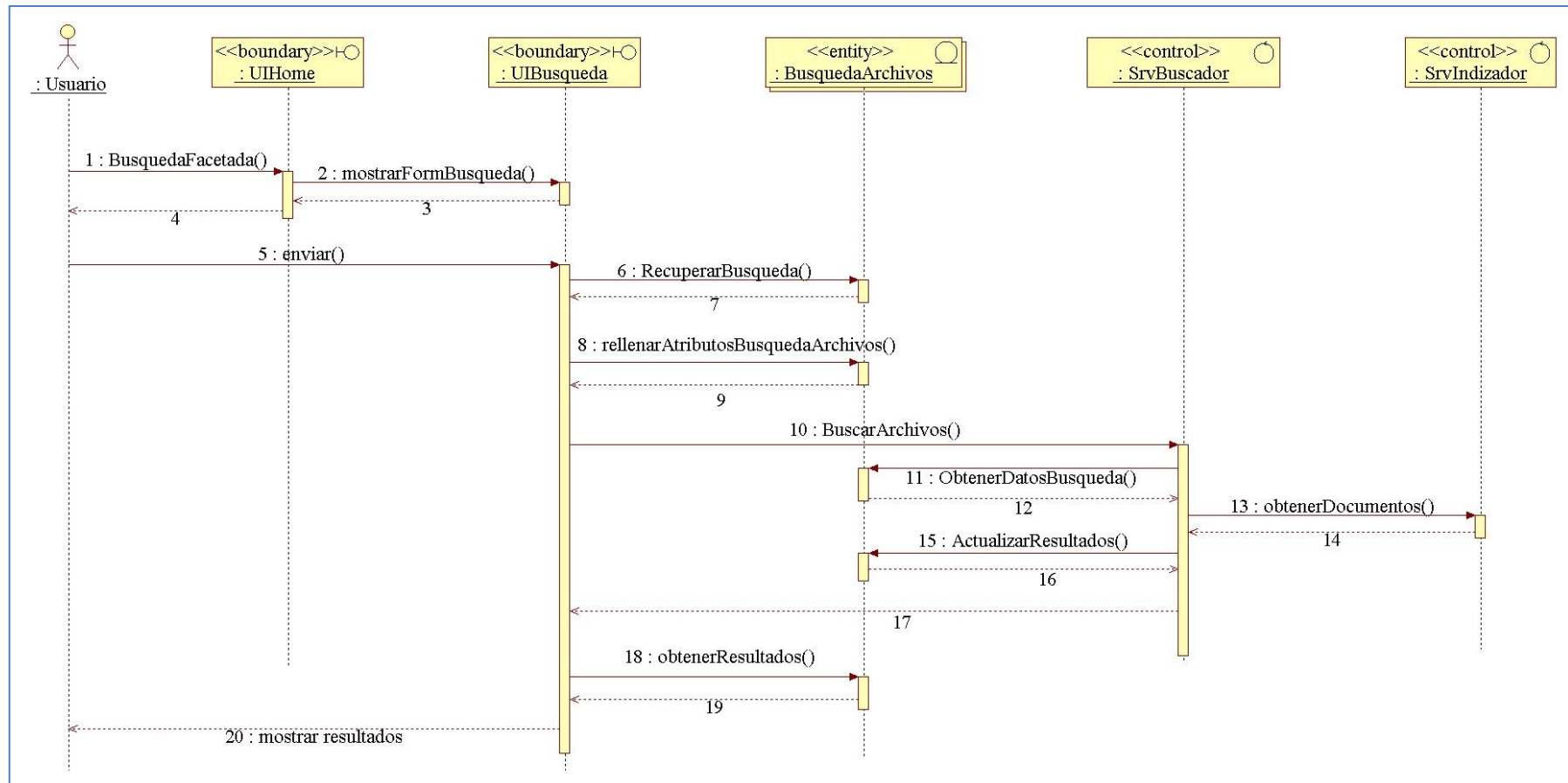


Ilustración 18- Diagrama secuencia de la búsqueda facetada

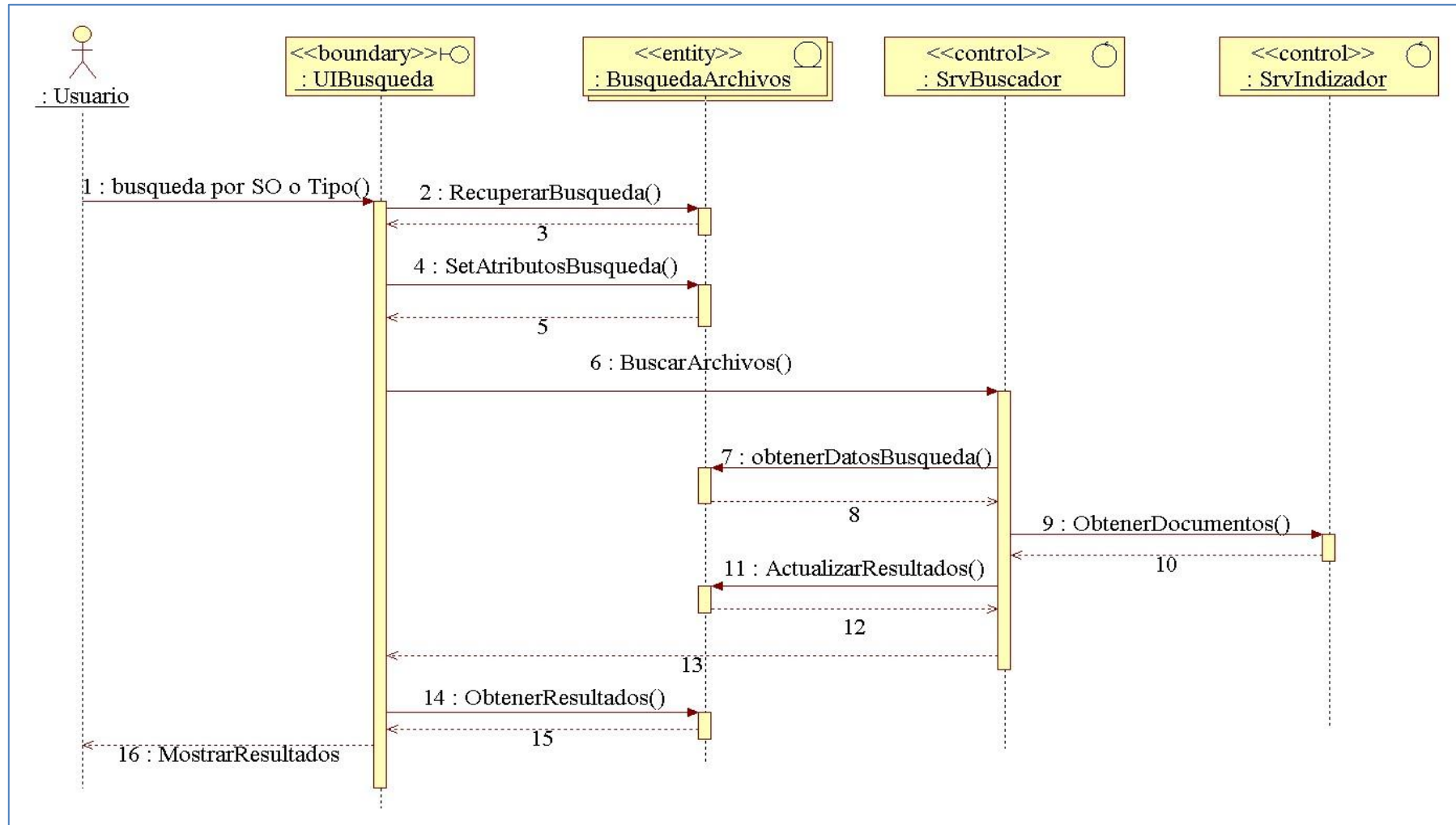


Ilustración 19- Diagrama de secuencia para la Búsqueda Por Sistema Operativo

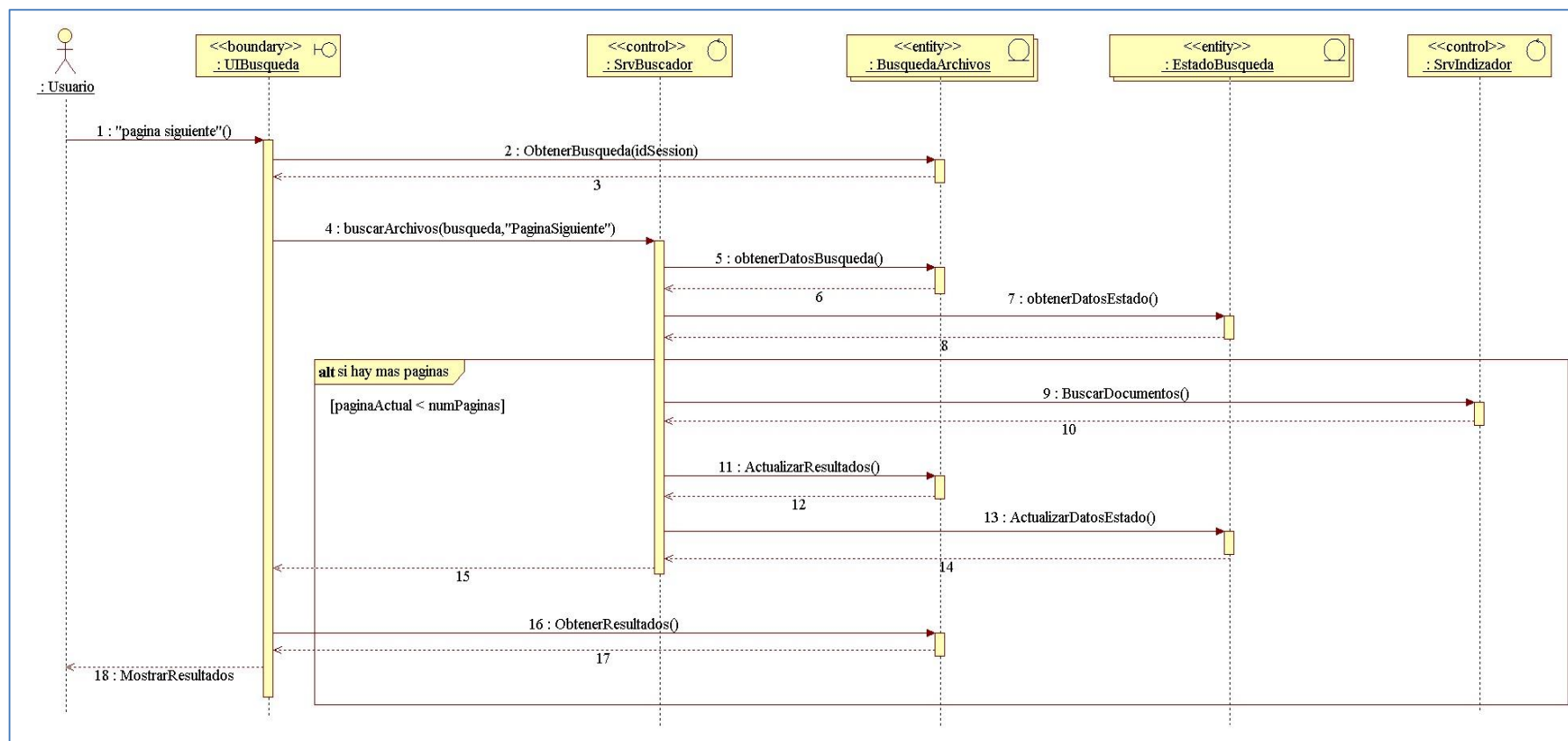


Ilustración 20- Diagrama de secuencia para mostrar página siguiente

5.3.2 Modelo de diseño para C2-VerArchivos

El caso de uso C2-VerArchivos se documenta en un solo diagrama. El usuario elige entre ver archivos de las listas de los más frecuentemente descargados, de la lista de los más recientes o bien realiza una búsqueda y escoge un archivo para ver sus propiedades.

En ambos casos se piden los datos del archivo a la base de datos. Si el archivo no está en memoria se crea la clase correspondiente. Si el archivo ya estaba en memoria no hay que pedirlo a la base de datos y se utiliza la clase existente, con el consiguiente ahorro de tiempo. Esta operación se implementara mediante un framework de acceso a base de datos, que nos facilita las ventajas comentadas

Para ambas opciones es preciso recuperar los datos de comentarios de los usuarios al archivo, en este caso, por cada comentario se crearía una clase en memoria, siempre que no estuviera presente, de forma análoga a los datos del archivo.

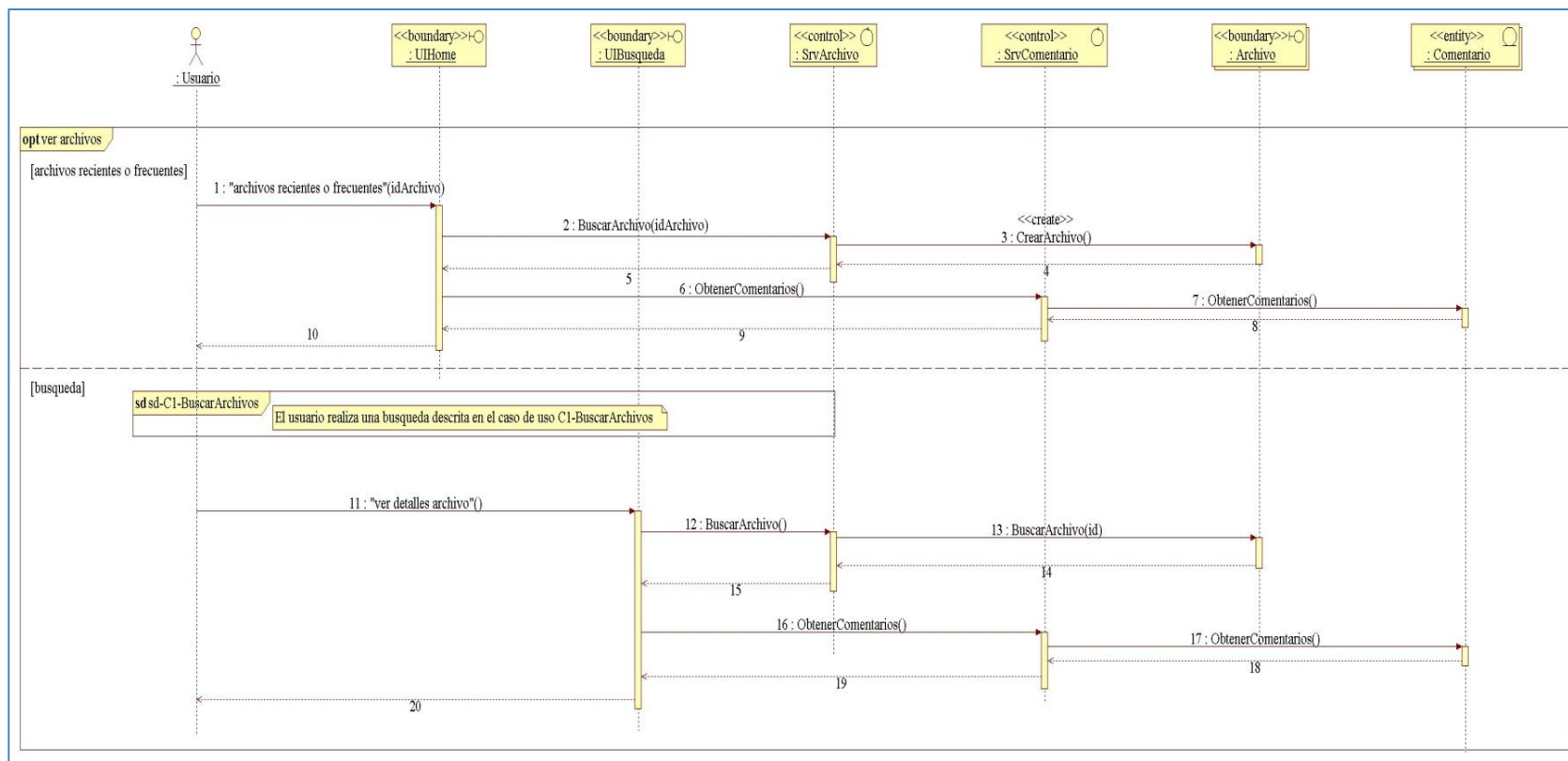


Ilustración 21- Diagrama de secuencia para el caso de uso C2-VerArchivos

5.3.3 Modelo de diseño para C3-DescargarArchivo

Para descargar un archivo el usuario puede escoger hacerlo desde dos sitios tal y como se indica en los requisitos:

- directamente desde la página de resultados de búsqueda o,
- desde la página de información de un archivo tras ver los detalles del mismo.

En el primer caso, recordamos que la precondition es que el usuario debe primero hacer una búsqueda fructífera. Para ello se hace una llamada al caso de uso C1-BuscarArchivos.

En el segundo caso, la precondition es que el usuario haga una búsqueda y después solicite ver los detalles de un archivo para que el sistema muestre la página de información del mismo. Por ello solo se hace una llamada al caso de uso C2-VerArchivos, ya que este caso de uso a su vez llama al caso C1-BuscarArchivos.

Necesitamos recuperar los datos del archivo para conocer la ruta física donde está alojado. Para ello se lo pedimos a la clase SrvArchivo. Solo se crea la clase del archivo si no estuviera ya en memoria. Una vez obtenida la ruta, las clases controladoras UIBusqueda o UIArchivo son las que realizan la descarga del archivo

Para este caso de uso se ha decidido que hasta que no se actualice la base de datos con el número de descargas, no se proceda a descargar el archivo.

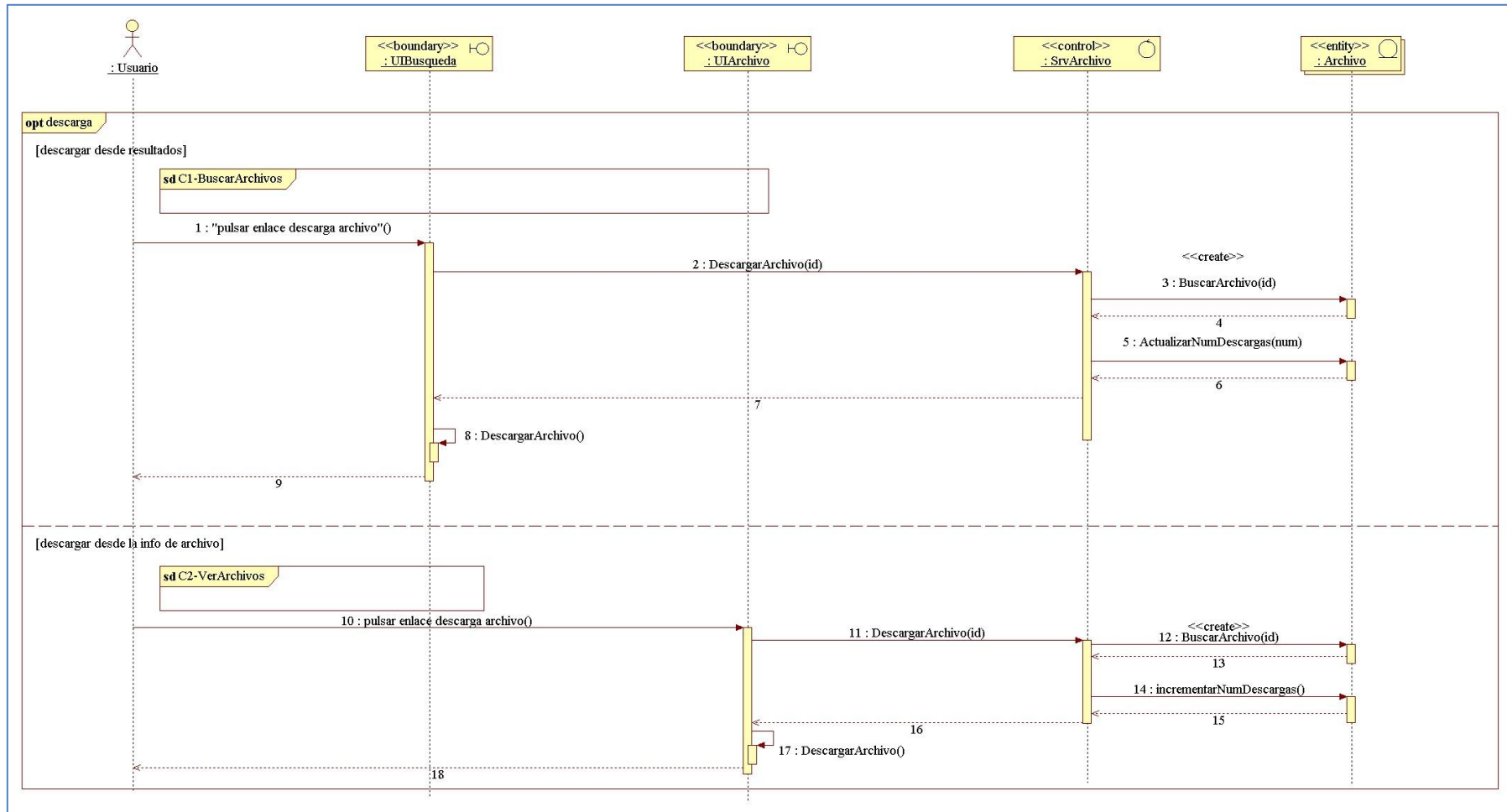


Ilustración 22- Diagrama de secuencia para el caso de uso C3-DescargarArchivo

5.3.4 Modelo de diseño para C4-Registrarse

Para este caso de uso se ha pensado que el usuario solicite a la interfaz principal un formulario para introducir los datos en el cual se mostrarían los posibles errores que cometa el usuario. La clase controladora UIRegistro invoca la clase de negocio SrvUsuario para comprobar que el alias no exista. Si es así se indica el error en el formulario. Una vez hecha la comprobación, la controladora crea la nueva clase de tipo Usuario, rellena los campos y vuelve a invocar a la clase de negocio para almacenar el usuario en base de datos.

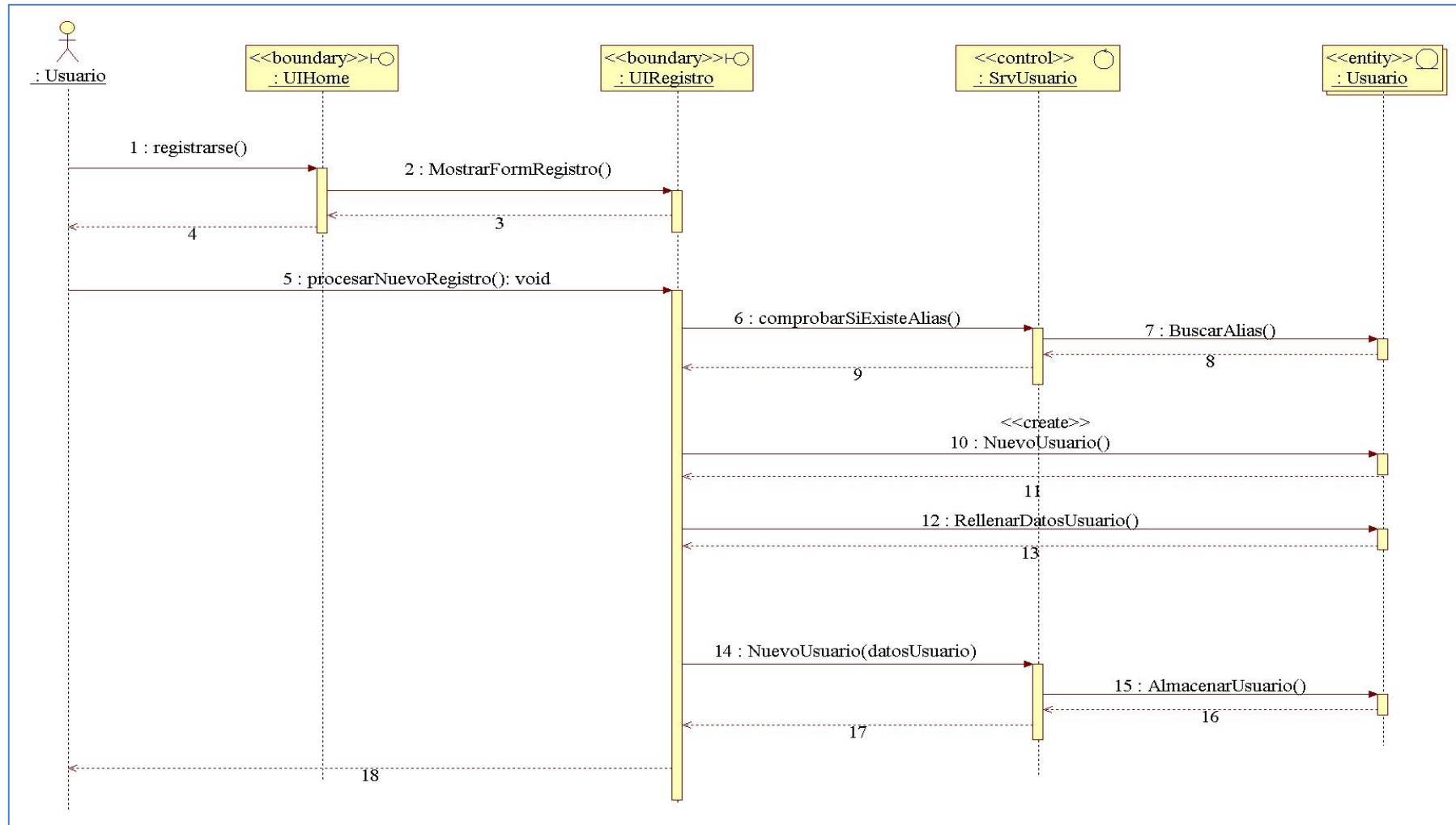


Ilustración 23- Diagrama de secuencia para el caso de uso C4-Registrarse

5.3.5 Modelo de diseño para C5-ModificarRegistro

Para este caso de uso se ha decidido que se comprueba la autenticación del usuario cuando éste envíe los datos, es decir que la opción estará disponible en la interfaz principal aunque no se haya autenticado

El usuario pide el formulario, rellena los datos, lo envía, y el sistema comprueba si está autenticado. Si no es así el sistema presenta el formulario de autenticación para que el usuario se autentique (C6-Autenticación). Cuando termina de hacerlo el sistema comprueba si ya se ha autenticado. Si ya lo ha hecho el sistema comprueba que los datos son correctos. Si es así procede a recuperar la clase Usuario correspondiente invocando a la clase de negocio, y la vuelve a invocar después de rellenar los nuevos datos para modificar el usuario en la base de datos.

Hemos decidido que el alias no se puede modificar, así que no hay que comprobar que éste no exista.

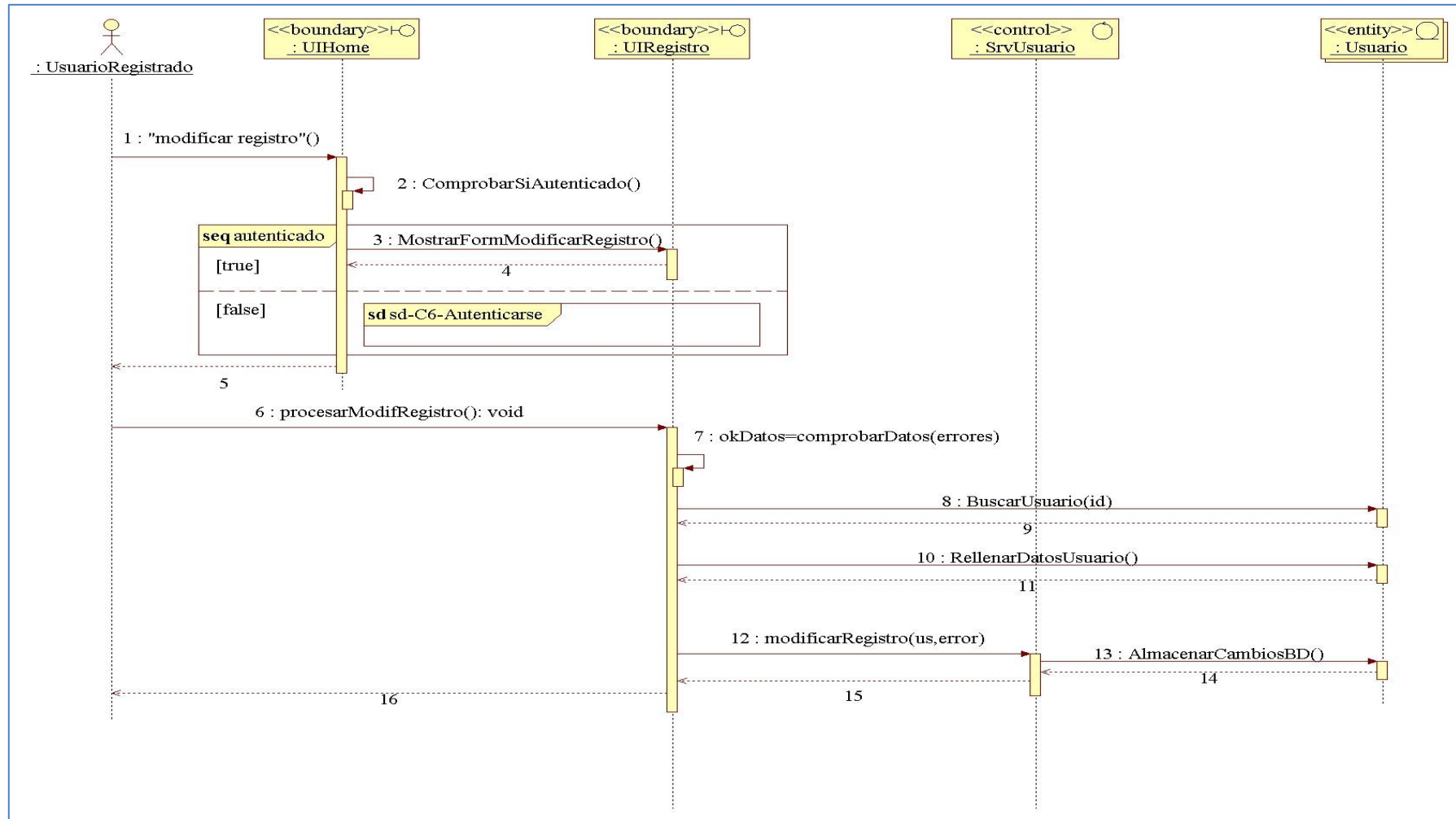


Ilustración 24- Diagrama de secuencia para el caso de uso C5-Modificar Registro

5.3.6 Modelo de diseño para C6-Autenticarse

Este caso de uso puede ser disparado de dos formas:

- por parte del usuario, cuando decide autenticarse y solicita el formulario o bien
- por parte del sistema, si es que el usuario solicita un servicio que requiera autenticación y dicho usuario no se haya autenticado previamente. En este caso será el sistema quien dispare el caso de uso mostrando el formulario de autenticación, para después volver al origen una vez autenticado el usuario.

Cuando el usuario manda los datos, la clase controladora UIAutenticacion invoca a la clase de negocio SrvUsuario para comprobar si son correctos el alias y la contraseña. La clase de negocio recupera la clase entidad de Usuario, por medio de un framework de base de datos, que si no existe en memoria principal la crea.

El sistema mostrará el formulario de autenticación mientras los datos no sean correctos.

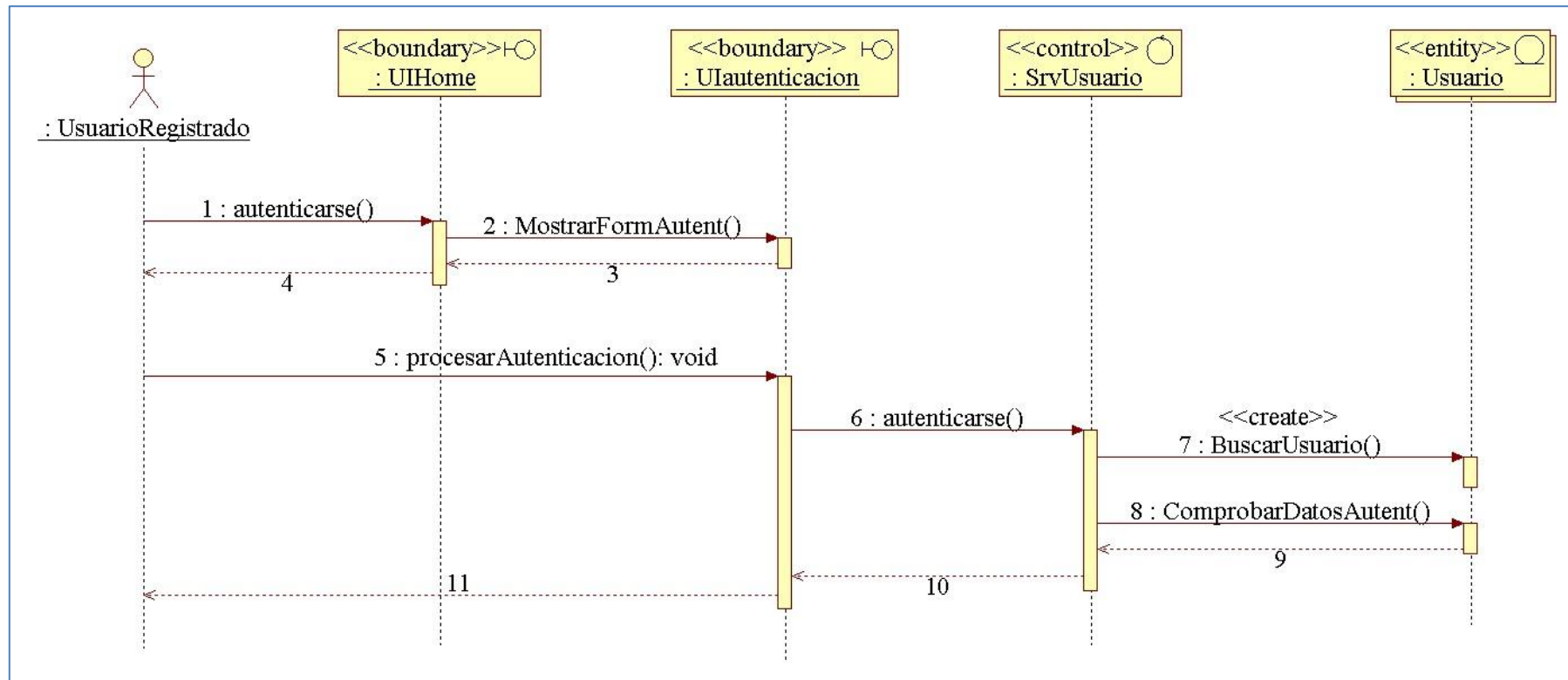


Ilustración 25- Diagrama de secuencia para el caso de uso C6-Registrarse

5.3.7 Modelo de diseño para C7-SubirArchivo

Para este caso de uso decidimos que cuando el usuario envía los datos se comprueba si está autenticado, si no lo está se ejecuta el caso de uso Autenticarse. La opción está presente en pantalla aunque el usuario no esté autenticado. La clase controladora UIArchivo solicita la ubicación a la clase de negocio SrvRepositorio. A continuación escribe el fichero en la ubicación recibida.

La siguiente operación por parte de la clase controladora es solicitar los detalles del archivo solicitándolo a los servicios de Windows. Una vez recopilados los datos procede a crear y rellenar una nueva clase Archivo, tras lo cual invoca a la clase de negocio SrvArchivo para almacenarla. La clase SrvArchivo almacena el archivo en base de datos por medio de un framework de base de datos, y a continuación invoca a la clase de negocio SrvIndizador quien indiza el archivo.

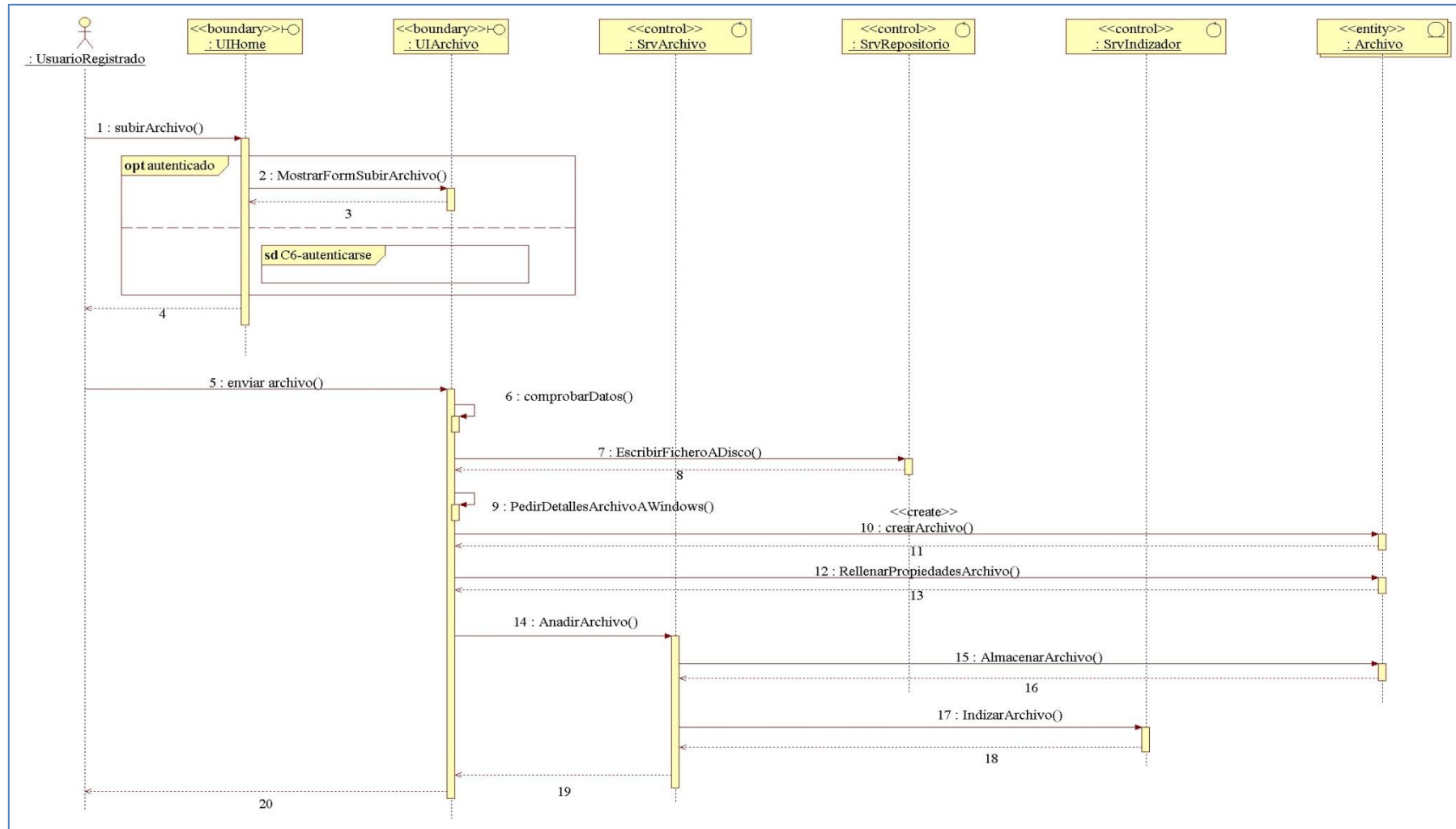


Ilustración 26- Diagrama de secuencia para el caso de uso C7-SubirArchivo

5.3.8 Modelo de diseño para C8-EliminarArchivo

Este caso de uso se plasma en un diagrama de secuencia. El prerequisite es haber solicitado ver los detalles de un archivo, como se ve en la llamada al caso de uso C2-VerArchivos. La página de detalles del archivo es el lugar desde el que un administrador podrá borrar el archivo

Es necesario estar autenticado como administrador ya que si no, no se mostrara el enlace para borrar el archivo en la página de detalles.

Se debería pedir confirmación antes de proceder al borrado. Si el usuario acepta se procede a borrar los archivos del almacenamiento físico, de memoria principal y de la base de datos. Para ello la clase controladora UIArchivo solicita borrar el archivo a la clase de negocio SrvArchivo que lo borrara de base de datos por medio de un framework de base de datos, del índice invocando a la clase de negocio SrvIndizador y físicamente invocando a la clase de negocio SrvRepositorio.

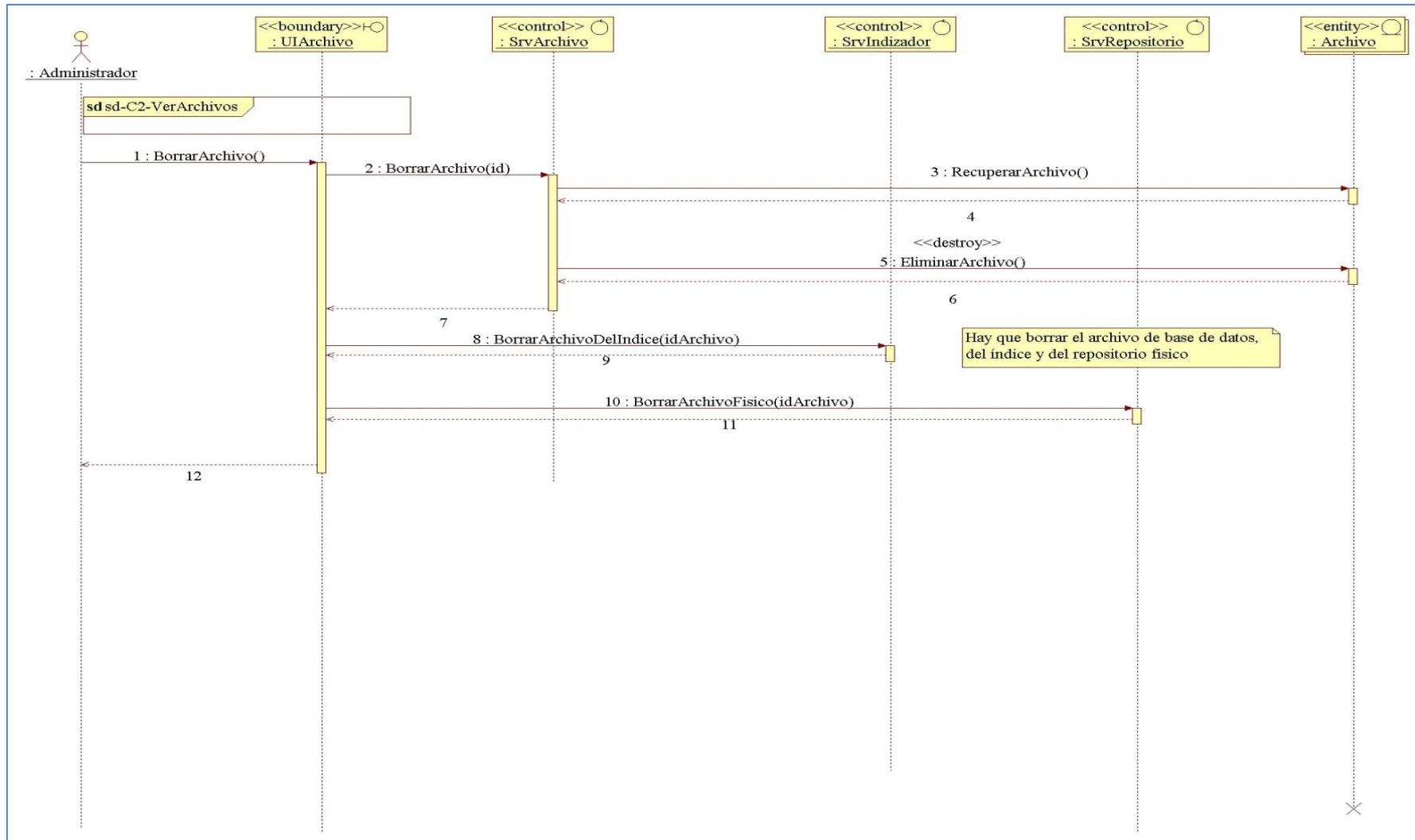


Ilustración 27- Diagrama de secuencia para el caso de uso C8-EliminarArchivo

5.3.9 Modelo de diseño para C9-ModificarArchivo

Para este caso de uso el prerequisite es haber solicitado ver los detalles de un archivo, como se ve en la llamada al caso de uso C2-VerArchivos. La página de detalles del archivo es el lugar desde el que un administrador podrá modificar el archivo

Es necesario estar autenticado como administrador ya que si no, no se mostrara el enlace para modificar el archivo en la página de detalles.

En primer lugar el usuario Administrador solicita modificar el archivo. La clase controladora UIArchivo mostrara el formulario para modificar los detalles del archivo. El usuario enviara los datos y la clase controladora comprobara los datos.

A continuación la clase controladora recuperara el archivo por medio de un framework de base de datos y modificara los campos del mismo, para después invocar a la clase de negocio SrvArchivo para que lo almacene en base de datos y lo modifique en el índice por medio de la clase de negocio SrvIndizador.

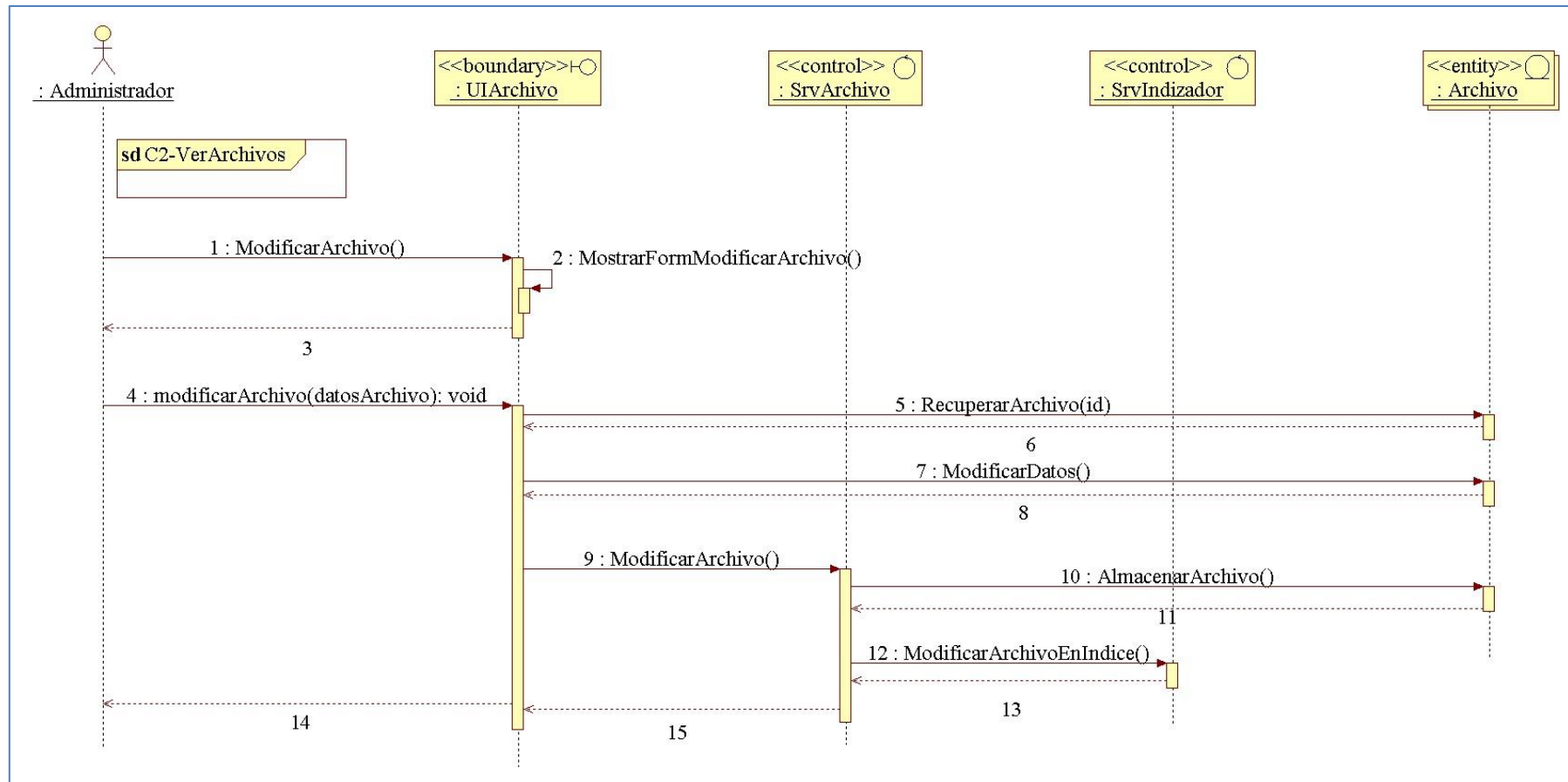


Ilustración 28- Diagrama de secuencia para el caso de uso C9-Modificar Archivo

5.3.10 Modelo de diseño para C10-SubirComentarioArchivo

En este caso de uso, de forma análoga a los dos anteriores, C8-EliminarArchivo y C9-ModificarArchivo, es preciso solicitar ver los detalles de un archivo por medio del caso de uso C2-VerArchivo. Es preciso estar autenticado ya que si no, no se mostraría el enlace para realizar la operación.

El usuario solicita comentar un archivo. La clase controladora UIComentario muestra el formulario. El usuario envía el formulario y la clase controladora los comprueba.

A continuación la clase controladora crea la nueva clase comentario y rellena sus campos con los datos. Llama a la clase de negocio SrvComentario enviándole los datos del comentario así como el id del archivo y el id del Usuario, para que los guarde en base de datos por medio de un framework de base de datos.

La clase de negocio almacena el nuevo comentario, y recupera todos los comentarios a ese archivo de base de datos para recalcular las valoraciones medias de ese archivo.

Una vez calculadas recupera el archivo, lo modifica y lo guarda en base de datos.

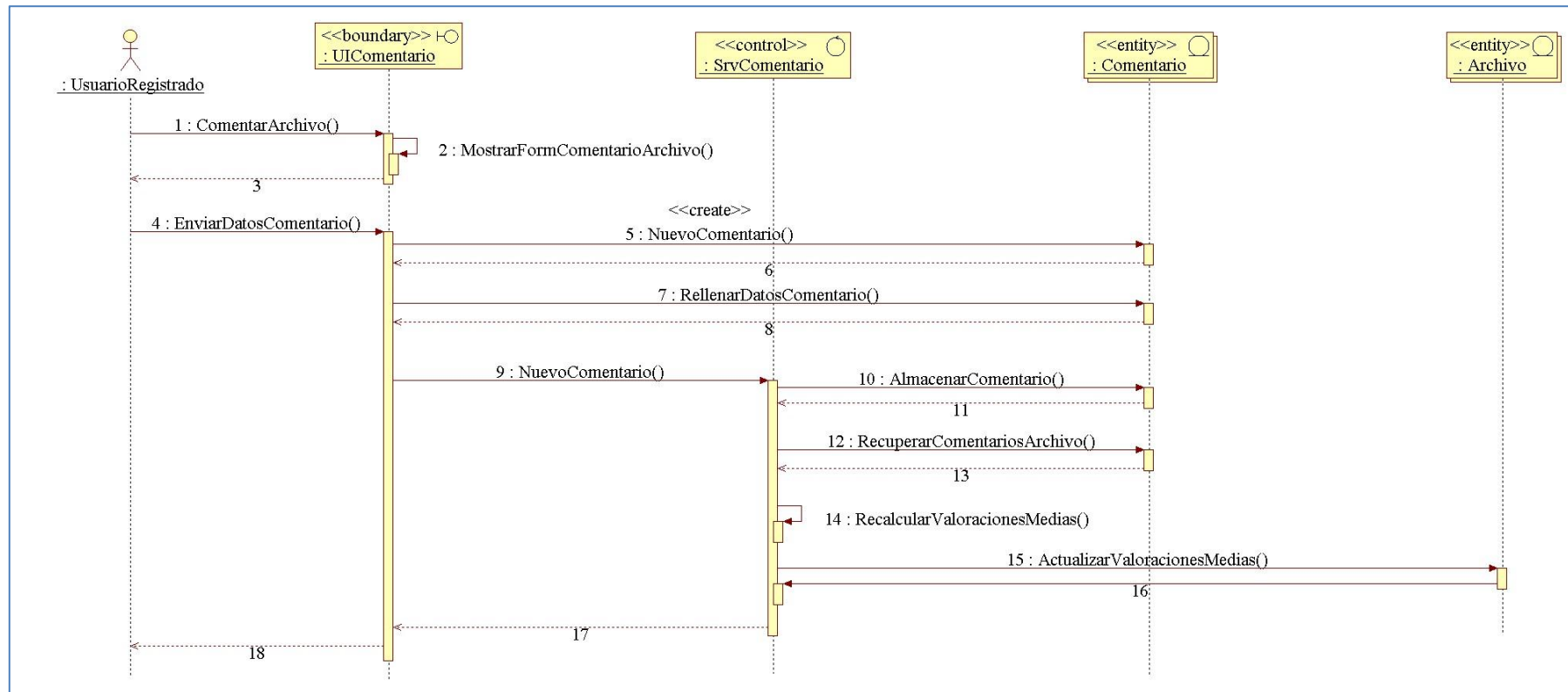


Ilustración 29- Diagrama de secuencia para C10-SubirComentarioArchivo

5.4 Arquitectura del software. Modelo de arquitectura

La arquitectura de un sistema de software es la parte responsable de identificar los componentes del sistema y sus relaciones para asegurar que el sistema satisface los requerimientos no funcionales [14]. En nuestro caso la arquitectura de nuestro de sistema está enfocada a conseguir que el sistema sea lo más **escalable** posible, es decir que se pueda adaptar a unas necesidades de servicio mayores que las actuales.

El marco en el que se encuadra el modelo arquitectónico es el paradigma **Modelo-Vista-Controlador (en adelante MVC)**

MVC es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y del módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el **modelo**, la **vista** y el **controlador**, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de la aplicación y su posterior mantenimiento.

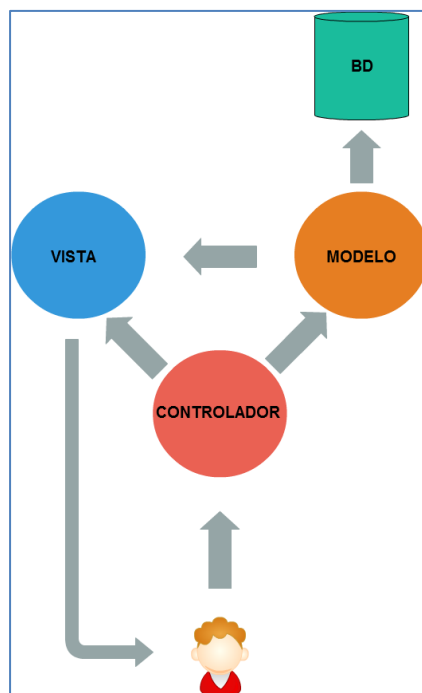


Ilustración 30- Esquema del patrón Modelo-vista-controlador

Según la metodología SunTone, un modelo de arquitectura típicamente incluye tres diagramas de despliegue:

- diagrama de despliegue de alto nivel
- diagrama de despliegue detallado.
- diagrama de paquetes de niveles y capas[14].

En nuestro caso la arquitectura está adaptada para un solo servidor con todo lo necesario ya que no se disponían de más medios a la hora de la implementación y pruebas, pero al ser una arquitectura de capas, estas capas se pueden separar e implementar en servidores diferentes. Así, el servidor de aplicaciones y el de base de datos podrían instalarse en otras máquinas diferentes e independientes del servidor web.

5.4.1 Diagrama de despliegue de alto nivel

Por medio de este diagrama se expone la arquitectura general del sistema. Muestra la distribución de los componentes de alto nivel de cada nodo hardware y como se despliegan esos componentes de software[14]. En nuestro caso es el siguiente:

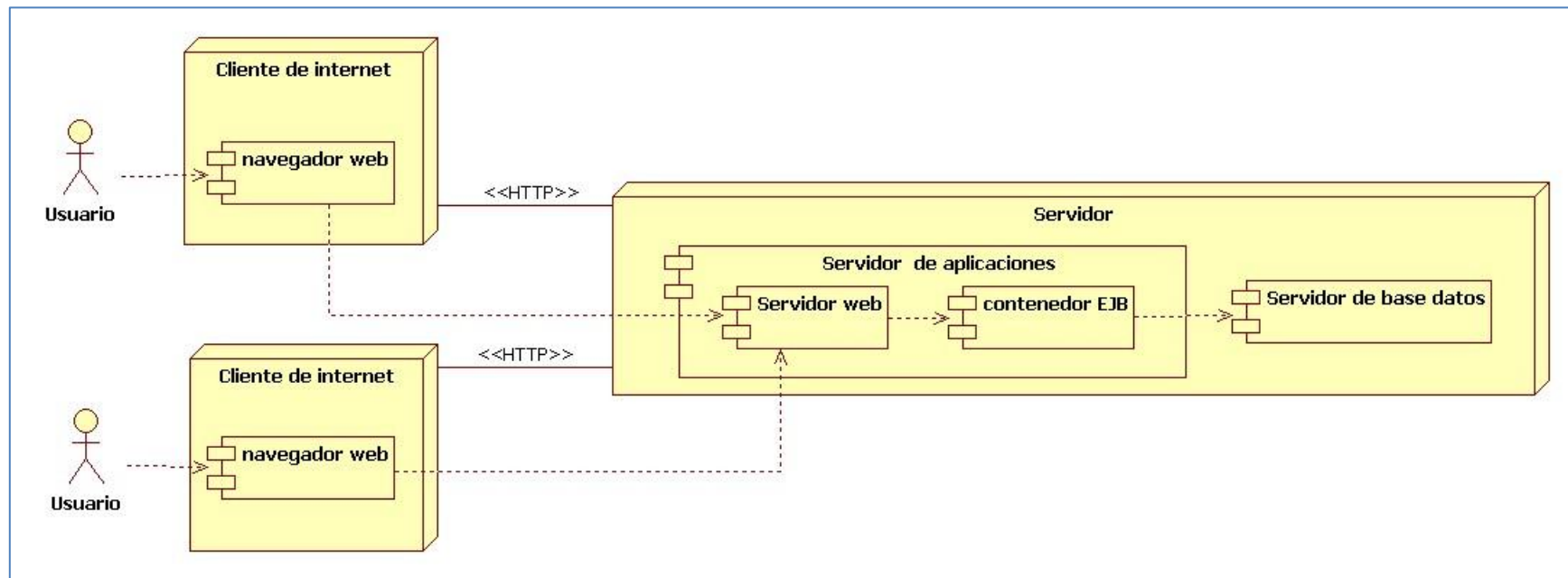


Ilustración 31- Diagrama de la arquitectura general del sistema

5.4.2 Diagrama de despliegue detallado

Un diagrama de despliegue detallado muestra los componentes que satisfacen uno o más casos de uso y como estos componentes se organizan dentro de la estructura de alto nivel del sistema[14].

Dicha organización que detallamos a continuación viene determinada por la aplicación del patrón MVC.

Modelo: nuestro modelo se compone de tres tipos de componentes relacionados entre sí:

- clases de servicio, que implementan la lógica de negocio.
- entidades persistentes, que mapean las entidades de base de datos y
- entidades no persistentes, que almacenan los datos de sesión de cada usuario: búsqueda y resultados

Las clases de servicio utilizan las entidades persistentes para recuperar y almacenar los datos en base de datos y las no persistentes para obtener los datos de sesión.

Controlador: los componentes de tipo controlador implementan las siguientes relaciones:

- invocan a las clases de servicio para obtener y almacenar los datos en base de datos
- crean clases de tipo entidad persistente
- modifican clases de tipo entidad no persistente
- recurren a un componente de tipo vista para presentar la información en pantalla.

Vista: un componente de tipo vista accede a un componente de tipo entidad para obtener los datos a presentar.

En nuestro caso vamos a presentar un diagrama general para todos los casos de uso, en el que de forma sencilla se muestran únicamente las capas de la arquitectura de la aplicación. En un modelo más exhaustivo tendríamos un diagrama por cada caso de uso en el cual estarían todos los componentes individuales que forman cada capa.

Como avanzábamos antes se ha decidido separar en servidores lógicos distintos la capa de presentación y la de negocio, para poder separar esta última y colocarla en un servidor distinto ante un hipotético crecimiento de demanda de servicio del sistema.

El diagrama es el siguiente:

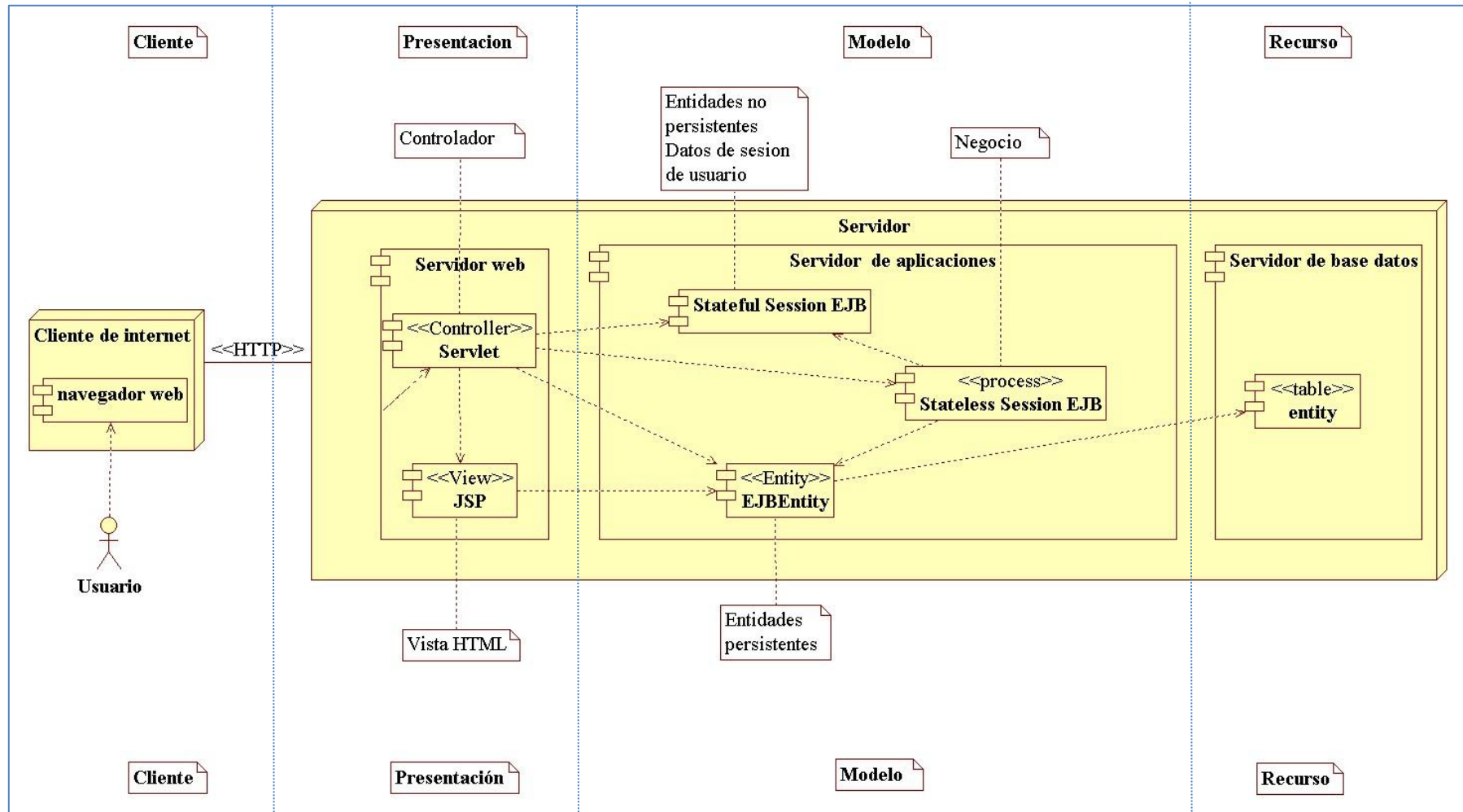


Ilustración 32- Diagrama de despliegue detallado del sistema

5.4.3 Diagrama de despliegue de niveles y capas

El diagrama de despliegue de niveles y capas muestra una matriz de paquetes en la cual se exponen las tecnologías seleccionadas por el responsable de la arquitectura del sistema [14]. En nuestro caso a nivel general se utiliza la tecnología JAVA, por requerimiento del tutor, y PostgreSQL, a elección nuestra.

Como decíamos el marco que guía la arquitectura del sistema es el patrón MVC. A continuación se expone la tecnología a utilizar en cada tipo de componente.

Vista

La vista se llevara a cabo mediante tecnología Java **JSP**. Para cada página web tendremos un archivo **jsp**.

Controlador

Los componentes de tipo controlador serán implementados por medio de **Servlets** Java.

Modelo

Para la capa de negocio se decide utilizar la tecnología **EJB** incorporando un contenedor de dicha tecnología independiente de la capa web que redunde como comentábamos en apartados anteriores en un sistema más escalable y que nos permitiría trasladar la capa de negocio a otro servidor, ante un importante aumento de demanda.

Otra ventaja que nos proporciona es la de que es accesible por cualquier cliente. Si en un futuro se quieren añadir otros tipos de clientes a la aplicación, como una aplicación de escritorio, la lógica de negocio seguiría siendo accesible.

Las clases de servicio se implementaran con EJB de tipo **Stateless Session**, (sesión sin estado).

Las clase de tipo entidad no persistente con EJB de tipo **Stateful Session** (sesión con estado).

Las clases de entidad persistente mediante EJB de tipo **Entity** (entidad).

Es en esta capa donde además se va a utilizar Lucene que nos va a facilitar la búsqueda facetada de archivos. Lucene es un motor de búsqueda utilizado para la clasificación e indexación de la información de los archivos.

En cuanto a la integración con la base de datos se decide utilizar el framework de control de persistencia JPA, por la mismas razones ya comentadas de escalabilidad. JPA mantiene una cache de objetos entidad en memoria principal cosa que permite servir los datos en menos tiempo y atender más peticiones.

En la capa de recurso se ha decidido utilizar el gestor de base de datos PostgreSQL ya que además de ser gratuito, creemos que funciona mejor con grandes cantidades de registros que otros gestores de base de datos de uso gratuito. Una vez más nos facilitaría la escalabilidad del sistema.

En cuanto al sistema operativo del servidor, obligatoriamente tiene que ser Windows ya que, como dijimos, un archivo DLL es exclusivo de Windows.

El diagrama de niveles y capas es el siguiente:

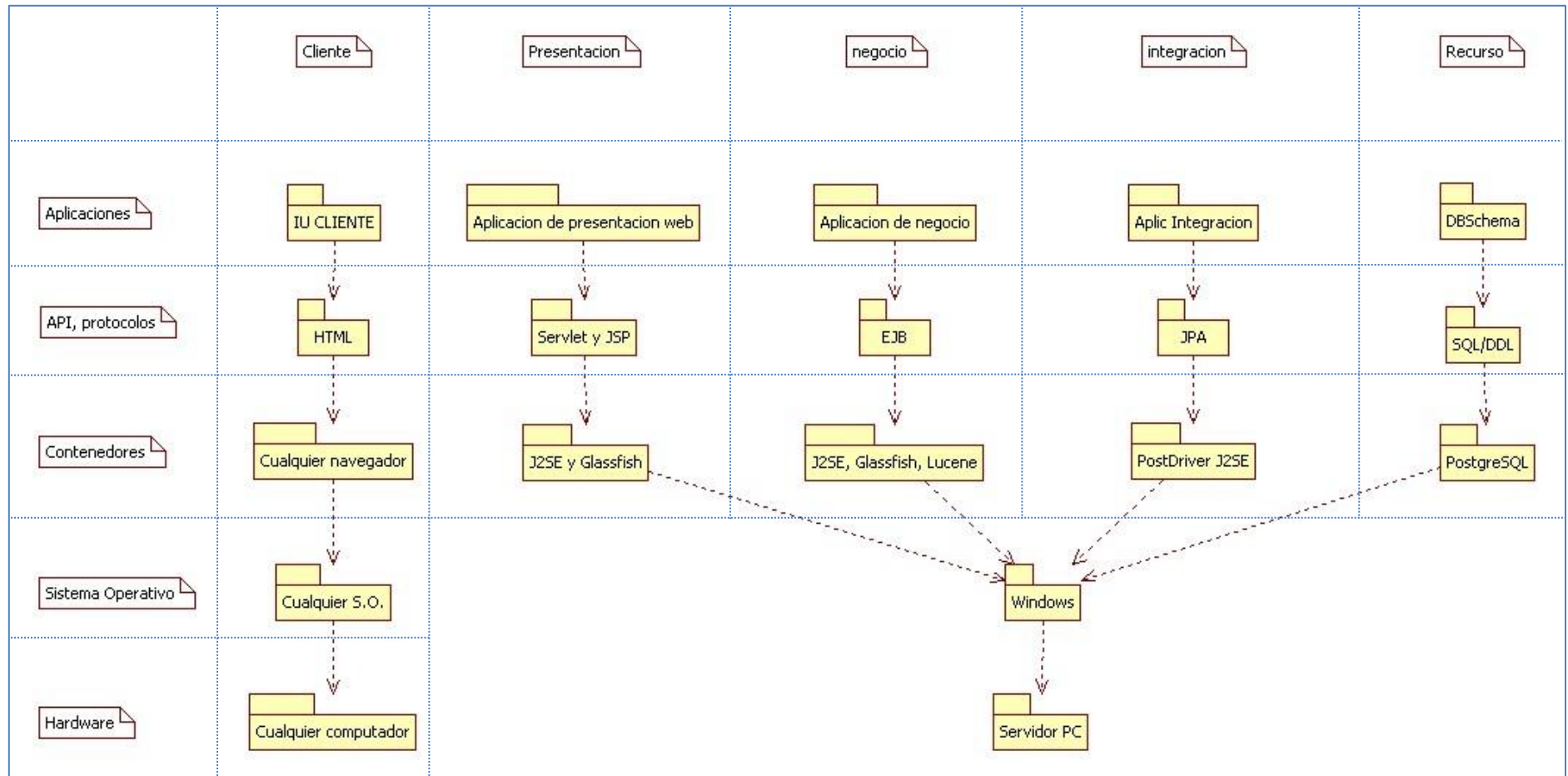


Ilustración 33- Diagrama niveles y capas del sistema

5.5 *Flujo de implementación*

En esta sección vamos a hablar de dos detalles fundamentales de implementación que son la indexación mediante Lucene como motor de búsqueda y la recuperación de información de versión de cada archivo por medio del API de Windows y la librería Jawin como framework para facilitar la tarea.

5.5.1 Indexación por medio de Lucene

Hasta ahora apenas hemos hablado de la indexación de la información. Necesitamos realizar o incorporar un software que nos facilite la búsqueda facetada de los archivos, mediante la indexación de la información de estos, de forma que mediante la creación y mantenimiento de un índice, a la hora de hacer una búsqueda el sistema nos devuelva instantáneamente una lista de resultados, en nuestro caso resultados relacionados con archivos.

Para ello como se indicaba en la arquitectura del sistema hemos decidido basar esta funcionalidad en el motor de búsqueda Lucene.

Lucene, de Apache, es un motor de búsqueda de texto completo de alto rendimiento escrito en su totalidad en java [15]. Es de código abierto y gratuito.

Nos facilita todo el análisis de texto y la creación y mantenimiento de un índice para búsquedas. Cuenta además con varios diccionarios para realizar el análisis cosa que aprovecharemos para incorporar los idiomas checo, alemán, griego, inglés, francés, holandés y portugués además del español en la clasificación del contenido y la búsqueda del mismo.

Lucene es simplemente un paquete de librerías java. Basta con incluirlo en nuestro proyecto e utilizarlo invocándolo desde nuestro código.

Con Lucene Tendremos que realizar dos operaciones:

- escritura del índice al insertar un nuevo archivo.
- lectura del índice al hacer una búsqueda.

Escritura del índice

Cada vez que se incluye un archivo nuevo en el repositorio y una vez obtenidos los detalles (facetar) del archivo, procederemos a invocar a Lucene para indexar la información aportada por el nuevo archivo incluido, que pasara a formar parte del índice que mantiene Lucene en la memoria. Si el lenguaje del archivo es uno de los que Lucene tiene disponibles lo indizaremos con el diccionario correspondiente. Si no es así se utilizara el diccionario inglés. Para ello seguimos los siguientes pasos:

- Crearemos un objeto IndexWriter (escritor de índice) dándole la ruta física donde tenemos alojado el índice de Lucene.
- Por medio del objeto IndexWriter crearemos un nuevo objeto Document (documento) por cada archivo a indexar y lo añadiremos mediante el método addDocument
- Por cada campo del archivo creamos un campo dentro del objeto Document (objeto Field o NumericField), analizando la información mediante un analizador correspondiente que será una clase hija de la clase Analyzer

Lectura de índice

Para la lectura del índice, utilizaremos el objeto IndexSearcher. En la búsqueda se dará opción a elegir el idioma del texto a escribir, y de nuevo mediante un objeto Analyzer correspondiente al idioma escogido se analiza el texto de búsqueda.

El resultado del análisis se envía al objeto IndexSearcher quien consulta el índice, y devuelve los resultados.

5.5.2 Invocación al API de Windows 32-bit por medio de Jawin

A continuación hablaremos de la parte que más ha costado desarrollar en este sistema. La invocación al API de Windows (escrito en C) desde nuestro código Java para obtener los datos de versión y detalles de los archivos.

Cada archivo contiene en su interior un bloque de datos de versión formado por una parte fija y otra variable, la parte fija contiene los datos de versión y la parte variable contiene los detalles. De la fija obtendremos los siguientes campos, todos ellos de cuatro bytes:

- dwSignature: campo de control, no lo vamos a indizar;
- dwStrucVersion: versión de la propia estructura, no lo vamos a indizar;
- dwFileVersionMS: parte más significativa de la versión del archivo, de ella obtenemos:
 - FileMinorVersion (2 bytes): número secundario, segundo número de los cuatro que forman la versión del archivo;
 - FileMajorVersion (2 bytes): número principal de la versión, el primer número de los cuatro que forman la versión;
- dwFileVersionLS: parte menos significativa del número de versión del archivo, de ella obtenemos:
 - FileRevisionNumber (2 bytes): número de revisión de la versión, el cuarto número de la versión;
 - FileBuildNumber (2bytes): número de compilación de la versión, tercer número de la versión;
- dwProductVersionMS: parte más significativa de la versión del producto al que pertenece el archivo, de ella obtenemos:
 - ProductMinorVersion (2 bytes): (número secundario, segundo número de los cuatro que forman la versión del producto);
 - ProductMajorVersion (2 bytes): (número principal de la versión, el primer número de los cuatro que forman la versión);
- dwProductVersionLS: parte menos significativa del número de versión del producto al que pertenece el archivo, de ella obtenemos :
 - ProductRevisionNumber (2 bytes): número de revisión de la versión, el cuarto número de la versión;
 - ProductBuildNumber (2bytes): número de compilación de la versión, tercer número de la versión;

- **dwFileFlagsMask**: mascara de bits que especifica que bits son válidos en el campo **dwFileFlags** tampoco vamos a indexar este campo;
- **dwFileFlags**: mascara de bits que especifica atributos booleanos del archivo. No se va a indexar este campo;
- **dwFileOS**: constante que representa el sistema operativo para el cual el fichero; fue diseñado.
- **dwFileType**: constante que representa el tipo de archivo;
- **dwFileSubtype**: constante que representa la función del archivo, solo nos interesa en caso de que el archivo sea Driver o Font, que nos dirá el subtipo;
- **dwFileDateMS**: parte más significativa de la fecha de creación del archivo;
- **dwFileDateLS**: parte menos significativa de la fecha de creación del archivo.

La parte variable puede contener una o varias hojas de detalles, cada una en un idioma y página de códigos. Cada hoja de detalles contiene los siguientes campos:

- **Lenguaje primario**: lenguaje en que está escrita la hoja de detalles.
- **SubLenguaje**: dialecto del lenguaje primario en que está escrita la hoja de detalles.
- **PaginaCodigos**: juego de caracteres en que está escrita la hoja de detalles
- **FileVersion**: número completo con las cuatro cifras separadas por puntos de la versión del archivo;
- **ProductVersion**: número completo con las cuatro cifras separadas por puntos de la versión del producto en que está incluido el archivo;
- **CompanyName**: organización que ha creado el archivo;
- **FileDescription**: descripción de las funciones del archivo;
- **InternalName**: nombre interno del archive;
- **LegalCopyright**: copyright del archivo;
- **LegalTrademarks**: marcas registradas del archivo;
- **OriginalFilename**: nombre original del archivo;
- **ProductName**: nombre del producto en el que está incluido el archivo.

El proyecto de integración Java/Win32 (Jawin) es una arquitectura de código abierto para la interoperación entre Java y los componentes expuestos a través del Component Object Model (COM) de Microsoft o a través de librerías de enlace dinámico (DLL).

Jawin permite a las aplicaciones Java llamar a cualquier código heredado basado en COM o en DLL sin tener que escribir ningún código en Java Native Interface (JNI), que es la manera estándar de acceder a código nativo desde la plataforma Java. JNI proporciona un conjunto de primitivas de bajo nivel que pueden ser usadas para aplanar parámetros desde java a código nativo y viceversa. JNI es una aproximación a muy bajo nivel para llamar a código nativo[16]. Por esa razón utilizaremos Jawin que nos facilitara esta tarea.

La DLL de Windows que nos proporciona los datos de versión de un archivo es **Version.dll**. Incluidas en ella están las funciones que nos interesan para obtener los datos incluidos dentro de cada archivo. Las funciones a invocar son:

- **GetFileVersionInfoSizeW**: nos devuelve el tamaño del bloque de datos de versión del archivo, dato necesario a la hora de recuperar dicho bloque puesto que el tamaño es variable en cada archivo.
- **GetFileVersionInfoW**: nos devuelve un array de bytes con el bloque completo de datos de versión.
- **VerQueryValueW**: nos devolverá un array de bytes con la parte del bloque que le indiquemos en cada momento mediante un parámetro de tipo cadena. Le proporcionaremos el bloque entero y de él nos extraerá lo que le indiquemos. Le podemos solicitar que nos devuelva la parte fija, los lenguajes en que tiene traducidos los detalles de versión, o un campo de una de las hojas de detalles de la parte variable.

El procedimiento a seguir es el siguiente:

1. Invocamos a **GetFileVersionInfoSizeW** para obtener el tamaño de la estructura.
2. Invocamos a **GetFileVersionInfoW** enviándole el tamaño para que nos devuelva la estructura completa.
3. Invocamos a **VerQueryValueW** para obtener la parte fija de la estructura de datos. Nos devuelve un array de bytes
4. Leemos del array los datos que nos interesan de la estructura saltándonos el resto.
5. Invocamos a **VerQueryValueW** para obtener los lenguajes y páginas de códigos presentes en el bloque.
6. Por cada lenguaje y página de códigos obtenemos los detalles de cada hoja realizando por cada campo una llamada a **VerQueryValue**.

5.6 *Flujo de implantación*

A continuación hablaremos de los requisitos que debe tener el entorno para que el sistema funcione correctamente, los ficheros de instalación de los que consta, el procedimiento para ponerlo en marcha y los archivos de configuración que habría que modificar si la nueva instalación tuviera otros datos distintos a los actuales.

5.6.1 Requisitos

El sistema se ha desarrollado para un entorno con los siguientes requisitos:

- Java JRE 1.7: entorno de ejecución de java.
- GlassFish Server 4: servidor de aplicaciones.
- PostgreSQL 9.2: servidor de base de datos.

Si falta alguno no funcionaria, y si se instalan otras versiones no se podría garantizar su correcto funcionamiento.

5.6.2 Ficheros de instalación

La aplicación se compone de los siguientes ficheros:

- **PFC_repositorioDLL.ear**: archivo con extensión EAR (Enterprise Archive) con el sistema a desplegar en el servidor de aplicaciones, en nuestro caso Glassfish 4.
- **script_BD_RepositorioDLL.sql**: script de creación de la base de datos. Con él se crean las tablas con sus restricciones y se cargan los datos necesarios para empezar a funcionar.

5.6.3 Procedimiento

Estos son los pasos para instalar la aplicación:

1. Instalar el servidor de base de datos **Postgresql 9.2**.
2. Ejecutar el script **script_BD_RepositorioDLL.sql** para crear la base de datos en la consola de PostgreSQL.
3. Instalar el servidor de aplicaciones **Glassfish 4**.
4. Configurar el servidor Glassfish:
5. Desplegar el EAR de la aplicación en el servidor Glassfish.
6. Modificar archivos de configuración de la aplicación (opcional)

Configuración de Glassfish

Para configurar el servidor de aplicaciones seguiremos los siguientes pasos:

1. Nos conectamos a la consola de administración de Glassfish: en un navegador web escribimos la dirección <http://localhost:4848/> si la abrimos desde el servidor. Si no es así habría que sustituir localhost por el nombre del host donde tuviéramos alojado el servidor de aplicaciones. El puerto por defecto al instalar Glassfish es el 4848. Si no fuera este habría que sustituirlo en la url.
2. Crear un **pool** de conexiones de base de datos: Para ello:
 - i. En el menú principal, a la izquierda, pulsamos la opción *Recursos -> JDBC -> Pools de conexiones JDBC*.
 - ii. Pulsamos el botón *Nuevo*. En el formulario introducimos los siguientes datos:
 1. *Nombre de Pool*. Escribimos **connectionPool**
 2. *Tipo de recurso*. Escogemos **javax.sql.DataSource**
 3. *Proveedor de Controladores de la Base de Datos*. Escribimos **org.postgresql.ds.PGSimpleDataSource**
 4. En la propiedad *introspección* y en los bloques *Configuración de pool* y *Transacción* dejamos los valores por defecto.
 5. En Propiedades Adicionales rellenamos las siguientes:

- a. *URL* ->
jdbc:postgresql://localhost:5432/repositorio_archivos
 - b. *DriverClass* -> **org.postgresql.Driver**
 - c. *Port number* -> **5432** (puerto por defecto de PostgreSQL)
 - d. *Database name* -> **repositorio_archivos** (el nombre de la base de datos)
 - e. *User* -> **postgres** (usuario de conexión a la base de datos)
 - f. *Password* -> **postgres** (password del usuario de conexión)
 - g. *Servename* -> **localhost** (host donde tenemos el servidor de base de datos, si estuviera en una maquina distinta aquí iría el nombre de la misma)
3. Crear un **recurso JDBC** al que le asociaremos el pool creado en el paso anterior
- i. Pulsamos en el menú *JDBC -> Recursos de JDBC*
 - ii. Pulsamos en el botón Nuevo.
 - iii. En el formulario rellenamos los siguientes datos.
 1. *Nombre JNDI* -> **jdbc/repositorioArchivos** que es el identificador del recurso
 2. *Nombre de pool* -> **connectionPool** (pool que hemos creado antes y que le asociamos ahora).

Despliegue del archivo EAR

Para desplegar el EAR de la aplicación en el servidor Glassfish seguimos los siguientes pasos:

1. Abrimos la consola de administración de Glassfish. En un navegador web escribimos la dirección <http://localhost:4848/> si la abrimos desde el servidor. Si no es

así habría que sustituir localhost por el nombre del host donde tuviéramos alojado el servidor.

- a. En menú principal en la columna izquierda pulsamos en la opción *Aplicaciones*.
- b. En la página que nos aparece escogemos la opción *Archivo Empaquetado que cargar en el servidor* y pulsamos el botón *Desplegar*.
- c. Se nos abre una vista de ficheros, seleccionamos el archivo *PFC_repositorioDLL.ear*.
- d. Pulsamos aceptar.

5.6.4 Archivos de configuración de la aplicación

En los archivos de configuración opcionalmente tendríamos que configurar la ruta del repositorio físico si queremos que la ruta sea otra, y los datos de acceso a la base de datos si estos fueran distintos en la nueva instalación.

Ruta del repositorio físico

Es la ruta raíz del disco duro de donde cuelga el repositorio físico.

La ruta del repositorio físico está en el archivo **web.xml**. Dentro del EAR el archivo web.xml está en la ruta relativa siguiente:

PFC-war|web|WEB-INF

Por defecto la ruta del repositorio físico es *e:\repositorioDLL*

Si en la nueva máquina no existe la unidad E o deseamos poner otra ruta es este archivo donde habría que cambiarla.

Datos de configuración de la base de datos

Si los datos de nuestra instalación no coincidieran con los que hemos dicho en el apartado anterior debemos cambiarlo en el fichero ***glassfish-resources.xml***

La ruta relativa dentro del EAR es la siguiente:

PFC-ejb\setup\glassfish-resources.xml

En él se configuran el recurso JDBC y el pool de conexiones. Dentro del pool tenemos la configuración de las propiedades de base de datos:

- serverName -> **localhost** (nombre de la maquina donde está el servidor PostgreSQL)
- portNumber -> **5432**
- URL -> **jdbc:postgresql://localhost:5432/repositorio_archivos**
- User -> **postgres** (usuario de conexión a la base de datos)
- Password -> **postgres** (password del usuario de conexión a la base de datos).

Si los datos son otros es en este archivo donde los cambiaremos.

6. Fase de Transición. Evaluación

En la fase de transición vamos a proceder a evaluar el rendimiento de nuestro producto. Vamos a verificar hasta qué punto la aplicación es escalable, es decir que va a seguir funcionando con un rendimiento aceptable aunque aumenten el número de usuarios, siempre y cuando el hardware de la maquina donde la instalemos nos lo permita.

6.1 Herramientas y condiciones

Para realizar las pruebas de carga o rendimiento utilizaremos la herramienta **Jmeter** de Apache. Con ella vamos a simular un escenario de uso lo más real posible en el cual varios usuarios realizan cosas distintas al mismo tiempo interactuando con la aplicación.

Jmeter nos permite añadir varios grupos de hilos distintos. Un grupo de hilos es un conjunto de usuarios realizando la misma actividad. Con cada grupo de hilos vamos a implementar un caso de uso, en cada uno se añaden las urls correspondientes al recorrido del escenario. En cada hilo podemos configurar cuantos usuarios lo realizan al mismo tiempo y cuantas veces repiten la actividad. En cuanto a este último dato, **100** repeticiones nos parece una cantidad adecuada para cada grupo de hilos.

Jmeter nos permite además facilitar un fichero de texto como entrada para cada grupo de hilos. Gracias ello hemos podido facilitar datos de prueba distintos para cada repetición del bucle [17].

Las pruebas se realizan con un ordenador portátil **con 2 Gigabytes de RAM** y un procesador Dual Core de **1 GigaHercio**.

Hay que resaltar que todo se ejecuta en una sola máquina: servidor de aplicaciones, servidor de base de datos, repositorio de archivos, y por último el soporte físico del índice, que para evitar reconstruirlo cada vez se escribe en disco.

Partimos de una base de datos de unos **5000** archivos DLL, aunque aumentará durante el proceso ya que uno de los casos de uso consiste en subir nuevos archivos.

6.2 *Escenario*

La descripción de los grupos de hilos con el escenario a simular es la siguiente:

1. Usuarios anónimos solicitan y descargan un archivo de la lista de los más descargados.
 - a. Home – el usuario solicita la página Home.
 - b. Ver Archivo – el usuario pulsa en uno de los archivos de la lista de los más descargados y solicita la vista de archivo.
 - c. Descargar Archivo – el usuario pulsa en descargar.
2. Usuarios anónimos solicitan y descargan un archivo de la lista de los más recientes.
 - a. Home – el usuario solicita la página Home.
 - b. Ver Archivo – el usuario pulsa en uno de los archivos de la lista de los más recientes y solicita la vista de archivo.
 - c. Descargar Archivo – el usuario pulsa en descargar.
3. Usuarios anónimos realizan una búsqueda de texto básica, y descargan un archivo de los resultados.
 - a. Home - el usuario solicita la página Home.
 - b. Búsqueda por palabras – el usuario rellena la caja de texto de la búsqueda y pulsa enviar.
 - c. Ver archivo - el usuario pulsa en uno
4. Usuarios anónimos listan archivos por tipo y descargan un archivo de los resultados.
 - a. Home - el usuario solicita la página Home.
 - b. Clasificación por tipo – el usuario solicita la clasificación por tipos. Se muestra la lista de tipos.
 - c. Ver tipo – el usuario pulsa en un tipo.
 - d. Ver archivo - el usuario pulsa en uno de los archivos y se muestra la vista de archivo.
 - e. Descargar archivo – el usuario pulsa en descargar.

5. Usuarios anónimos listan archivos por SO, y descargan un archivo de los resultados.
 - a. Home - el usuario solicita la página Home.
 - b. Clasificación por SO – el usuario solicita la clasificación por Sistema Operativo. Se muestra la lista de sistemas operativos.
 - c. Ver SO – el usuario pulsa en un sistema operativo.
 - d. Ver archivo - el usuario pulsa en uno de los archivos y se muestra la vista de archivo.
 - e. Descargar archivo – el usuario pulsa en descargar.

6. Usuarios anónimos realizan una búsqueda Avanzada, y descargan un archivo de los resultados.
 - a. Home - el usuario solicita la página Home
 - b. Búsqueda avanzada mostrar form. – el usuario solicita el formulario de búsqueda avanzada
 - c. Búsqueda avanzada – el usuario envía los parámetros de búsqueda.
 - d. Ver archivo - el usuario pulsa en uno de los archivos de los resultados y se muestra la vista de archivo.
 - e. Descargar archivo – el usuario pulsa en descargar.

7. Usuarios registrados suben nuevos archivos al sistema.
 - a. Home - el usuario solicita la página Home.
 - b. Identificarse – el usuario solicita el formulario de identificación.
 - c. Mandar credenciales – el usuario envía su login y contraseña.
 - d. Mostrar formulario subir archivo – el usuario solicita el formulario para subir archivo.
 - e. Subir archivo. – el usuario envía el nuevo archivo.

8. Usuarios registrados comentan archivos del sistema.
 - a. Home - el usuario solicita la página Home.
 - b. Identificarse – el usuario solicita el formulario de identificación.
 - c. Mandar credenciales – el usuario envía su login y contraseña.
 - d. Clasificación por SO. – el usuario solicita el listado de sistemas operativos.
 - e. Ver SO – el usuario solicita el sistema operativo Windows XP.
 - f. Ver Archivo – el usuario pulsa en uno de los archivos de los resultados.
 - g. Insertar comentario – el usuario solicita el formulario para insertar comentario.
 - h. Enviar comentario – el usuario envía el formulario.

La combinación de algunos o todos los grupos de hilos, con 1 o varios usuarios por grupo y 100 repeticiones como hemos dicho, nos dará nuestro escenario de uso para las pruebas de rendimiento.

6.3 Descripción de las pruebas

Las pruebas que vamos a realizar son las siguientes:

1. Un grupo de hilos con 1 usuario y 100 repeticiones.
Tomamos el grupo de hilos número 7 con un usuario y 100 repeticiones.
2. Dos grupos de hilos con 1 usuario y 100 repeticiones (2 usuarios al mismo tiempo)
Tomamos los grupos de hilos números 7 y 5 con un usuario y 100 repeticiones cada uno.
3. Cuatro grupos de hilos con 1 usuario y 100 repeticiones (4 usuarios al mismo tiempo)
Tomamos los grupos de hilos 7, 5, 3 y 1 con un usuario y 100 repeticiones cada uno.
4. Ocho grupos de hilos con 1 usuario y 100 repeticiones (8 usuarios al mismo tiempo)
Tomamos los 8 grupos de hilos con un usuario y 100 repeticiones cada uno.
5. Ocho grupos de hilos con 2 usuarios y 100 repeticiones (16 usuarios al mismo tiempo)
Tomamos los 8 grupos de hilos con dos usuarios y 100 repeticiones cada uno.
6. Ocho grupos de hilos con 3 usuarios y 100 repeticiones (24 usuarios al mismo tiempo)
Tomamos los 8 grupos de hilos con dos usuarios y 100 repeticiones cada uno.

6.4 Resultados de las pruebas

En los resultados mostraremos los siguientes datos para cada prueba:

- tiempo de respuesta: media, mediana, máximo y mínimo
- rendimiento medido en peticiones por segundo
- rendimiento medido en Kilobytes por segundo

La manera de mostrar dichos resultados será con una tabla que Jmeter denomina Informe Agregado y una gráfica con la evolución del tiempo de respuesta.

En el informe agregado, para simplificar los resultados jmeter nos permite agrupar las peticiones comunes de los distintos grupos de hilos. Solo mostraremos los resultados para cada tipo de petición distinta. Mostraremos una tabla donde se detallan los resultados para cada tipo de petición y el total de la prueba, con los siguientes datos:

- Etiqueta – la Etiqueta de la muestra.
- # Muestras – El número de muestras (peticiones) con la misma etiqueta.
- Media – El tiempo medio de respuesta de un conjunto de resultados
- Mediana – la mediana (tiempo en la mitad de un conjunto de resultados, 50% de las muestras no llevan más de este tiempo).
- Min – el menor tiempo para el conjunto de muestras con la misma etiqueta.
- Max - el mayor tiempo para el conjunto de muestras con la misma etiqueta.
- Rendimiento – medido en peticiones por segundo.
- Kb/sec – rendimiento medido en Kilobytes por segundo

En la gráfica con el tiempo de respuesta se muestran media, máximo y mínimo.

6.4.1 Prueba 1: 1 grupo de hilos con 1 usuario

Informe agregado:

Etiqueta	# Muestras	Media	Mediana	Mín	Máx	Rendimiento	Kb/sec
Petición http	100	161	164	91	179	0,6	7,4
Identificarse	100	4	5	4	8	0,6	5,1
Mandar credenciales	100	8	9	5	11	0,6	4,6
Mostrar form. subir archivo	100	7	8	5	10	0,6	5,3
Subir archivo	100	77	63	35	363	0,6	4,6
Total	500	52	9	4	363	2,8	26,9

Tabla 29- Resultados de la prueba de rendimiento 1

Grafica con el tiempo de respuesta:

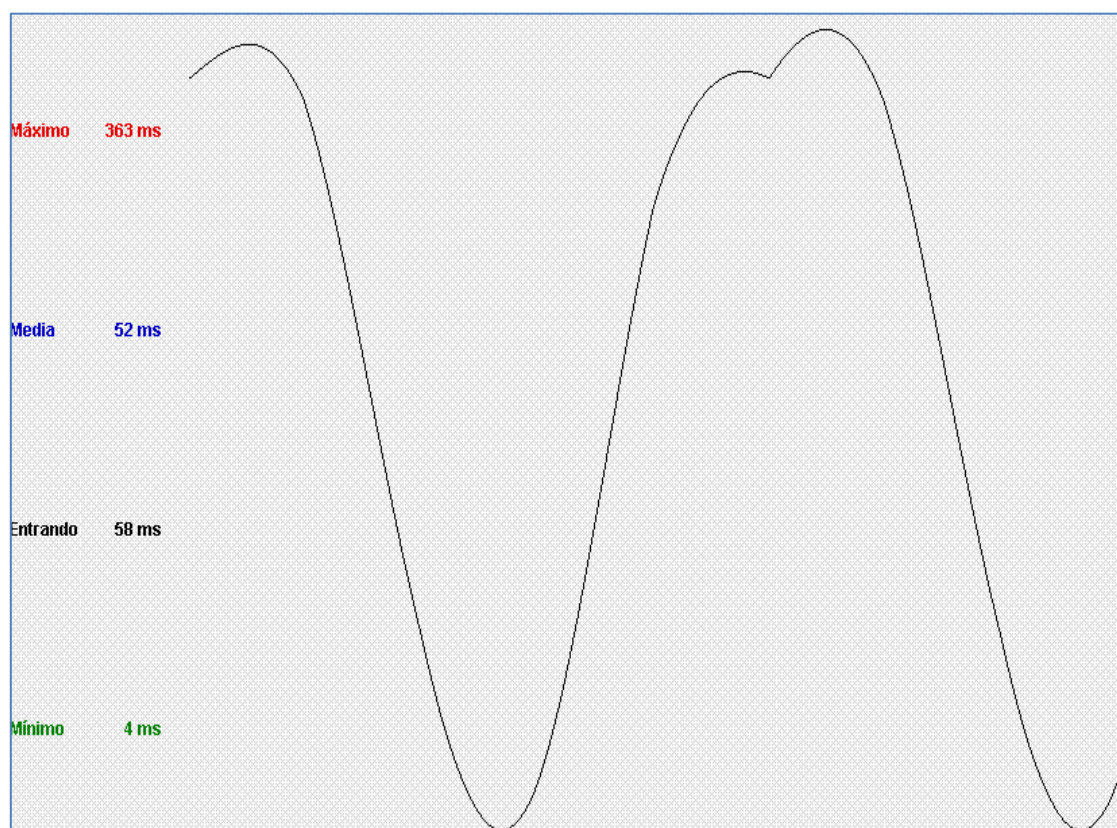


Ilustración 34- Gráfica del tiempo de respuesta de la prueba 1

6.4.2 Prueba 2: 2 grupos de hilos con 1 usuario por hilo

Informe agregado:

Etiqueta	# Muestras	Media	Mediana	Mín	Máx	Rendimiento	Kb/sec
Home	200	232	174	101	7452	1	12,9
Clasificación por so	100	15	5	4	1006	0,5	5,2
Identificarse	100	16	5	4	998	0,5	4,6
Mandar credenciales	100	16	10	6	560	0,5	4,2
Mostrar form subir archivo	100	20	8	5	1203	0,5	4,8
ver so	100	144	117	70	2461	0,5	15,4
Ver archivo	100	51	20	12	2971	0,5	8,7
Descargar archivo	100	40	8	4	651	0,5	74,3
Subir archivo	100	257	186	50	3841	0,5	4,2
Total	1000	102	19	4	7452	4,9	125,6

Tabla 30- Resultados de la prueba de rendimiento 2

Gráfica con el tiempo de respuesta:

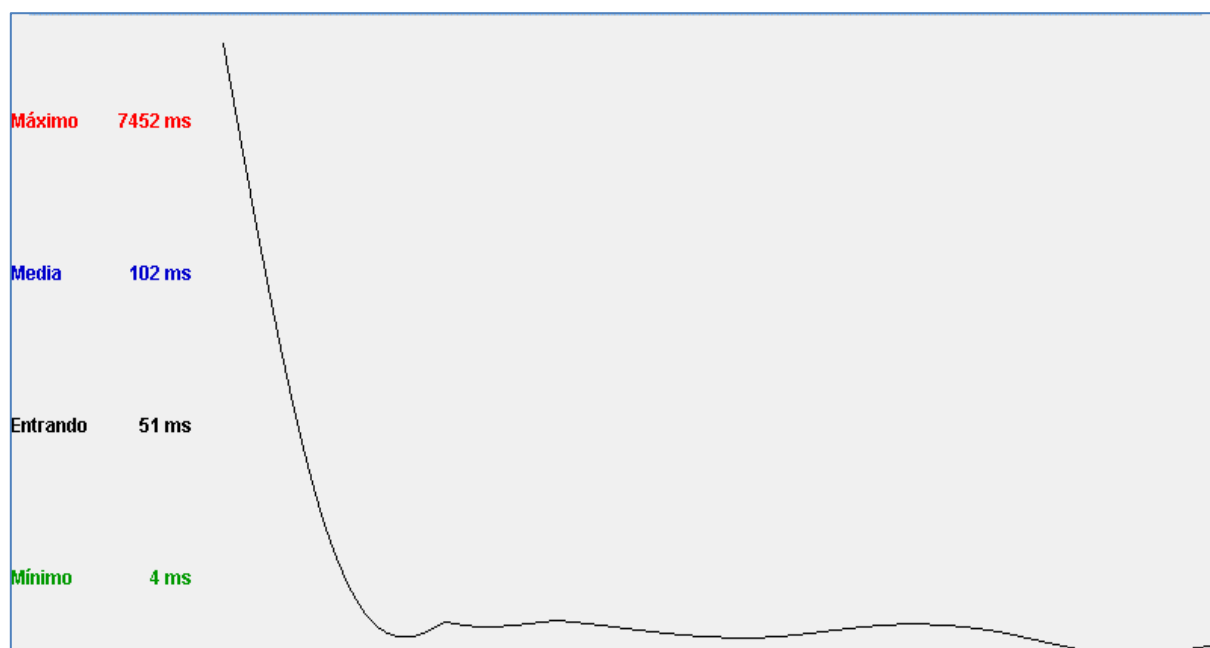


Ilustración 35- Gráfica del tiempo de respuesta de la prueba 2

6.4.3 Prueba 3: 4 grupos de hilos con 1 usuario por hilo

Informe agregado:

Etiqueta	# Muestras	Media	Mediana	Mín	Máx	Rendimiento	Kb/sec
Home	400	153	140	103	335	2,2	28,9
Identificarse	100	7	5	4	120	0,6	5,1
Ver archive	200	17	13	10	115	1,1	17,7
Clasificación por so	100	6	5	4	69	0,6	5,6
Mandar credenciales	100	10	8	4	77	0,6	4,6
Búsqueda por palabras	100	67	64	14	254	0,7	15,8
Descargar archive	300	20	8	3	182	1,7	280,5
ver so	100	108	110	58	283	0,6	16,6
Mostrar form subir archivo	100	9	7	4	79	0,6	5,3
Ver Archivo	100	19	17	9	139	0,7	9,4
Subir archive	100	68	55	29	297	0,6	4,6
Total	1700	59	19	3	335	9,3	387,3

Tabla 31- Resultados de la prueba de rendimiento 3

Grafica con el tiempo de respuesta:

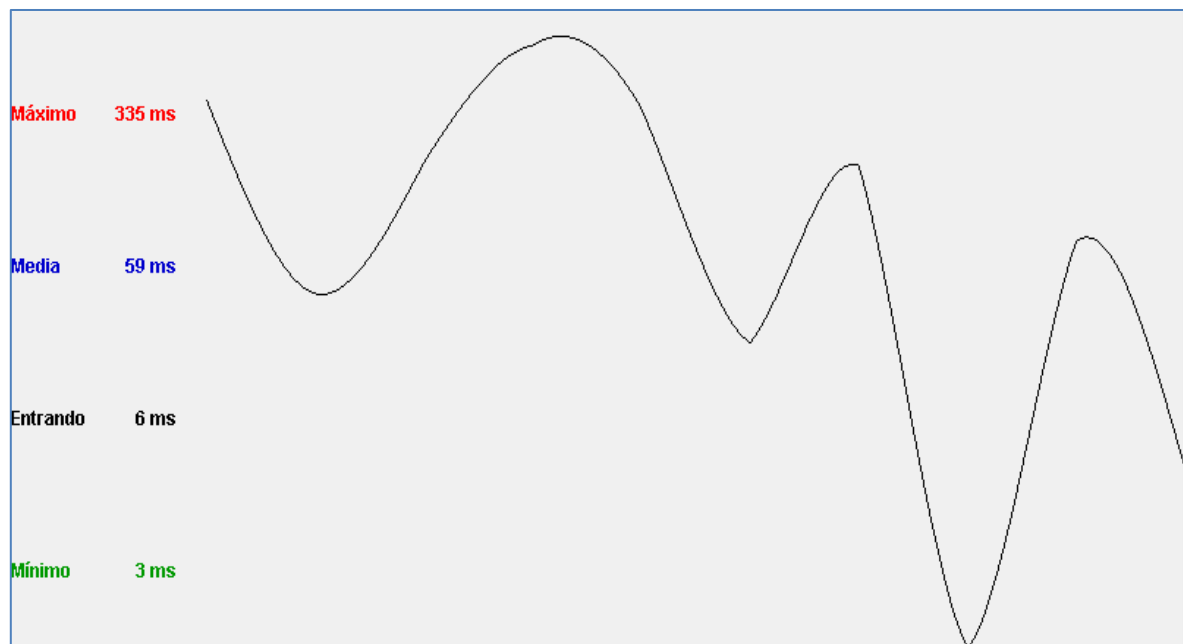


Ilustración 36- Gráfica del tiempo de respuesta de la prueba 3

6.4.4 Prueba 4: 8 grupos de hilos con 1 usuario por hilo

Informe agregado:

Etiqueta	# Muestras	Media	Mediana	Mín	Máx	Rendimiento	Kb/sec
Home	800	174	154	103	644	2,8	36,3
Ver archivo	400	29	14	9	347	2,1	35,2
Clasificación por so	200	18	7	3	284	0,7	7,1
Descargar archivo	400	27	8	2	317	2,1	342,9
Identificarse	200	17	5	3	269	0,7	6,3
Búsqueda por palabras	100	105	79	14	531	0,6	15,2
ver so	100	121	101	58	415	0,5	15,9
Descargar archivo	100	20	7	2	317	0,9	79,1
Mandar credenciales	200	14	7	4	139	0,7	5,7
Ver Archivo	300	30	19	9	292	1	20,5
Clasificación por tipo	100	28	7	4	956	0,5	6,1
Mostrar form subir archivo	100	16	7	4	183	0,5	5,1
ver tipo	100	111	92	59	468	0,5	16
Ver Windows xp	100	113	104	59	461	0,3	10,6
Búsqueda avanzada form	100	50	21	13	1230	0,6	68,7
Subir archivo	100	113	65	28	410	0,5	4,5
Búsqueda avanzada	100	133	94	30	424	0,6	15,4
Insertar comentario	100	31	8	4	841	0,3	4,5
Enviar comentario	100	49	28	14	757	0,3	9
Total	3700	73	29	2	1230	12,8	464,5

Tabla 32- Resultados de la prueba de rendimiento 4

Grafica con el tiempo de respuesta:

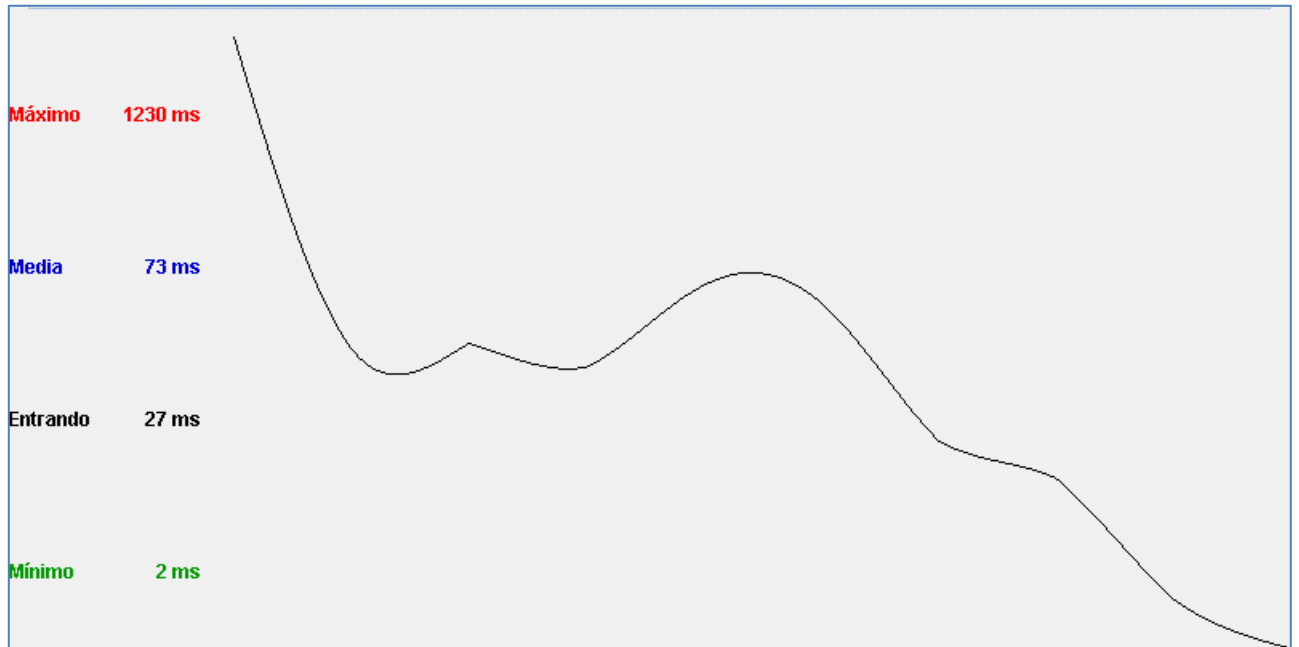


Ilustración 37- Gráfica del tiempo de respuesta de la prueba 4

6.4.5 Prueba 5: 8 grupos de hilos con 2 usuarios por hilo

Informe agregado:

Etiqueta	# Muestras	Media	Mediana	Mín	Máy	Rendimiento	Kb/sec
Home	1600	217	194	102	879	5,1	66,9
Identificarse	400	46	9	3	367	1,3	11,6
Ver archivo	800	80	39	8	662	3,7	62,4
Clasificación por tipo	200	54	20	4	447	0,9	10,7
Clasificación por so	400	55	11	3	752	1,3	13
Búsqueda avanzada form	200	131	45	12	1130	1,1	115,9
Mandar credenciales	400	42	9	3	308	1,3	10,5
Búsqueda por palabras	200	203	160	15	849	1,1	25,4
Descargar archivo	200	75	36	2	426	1,5	131,3
Ver Archivo	600	75	35	8	738	1,9	37,6
ver tipo	200	181	146	58	782	0,9	28,1
Descargar archivo	800	84	35	2	867	3,7	597,5
ver so	200	211	174	58	776	0,9	27,8
Búsqueda avanzada	200	214	186	28	802	1,1	25,8
Mostrar form subir archivo	200	60	18	3	379	0,9	8,9
Ver Windows xp	200	166	118	58	629	0,6	19,4
Subir archivo	200	194	150	27	817	0,9	7,8
Insertar comentario	200	47	9	4	399	0,6	8,3
Enviar comentario	200	76	29	14	492	0,6	16,5
Total	7400	122	94	2	1130	23,5	846,4

Tabla 33- Resultados de la prueba de rendimiento 5

Grafica con el tiempo de respuesta:

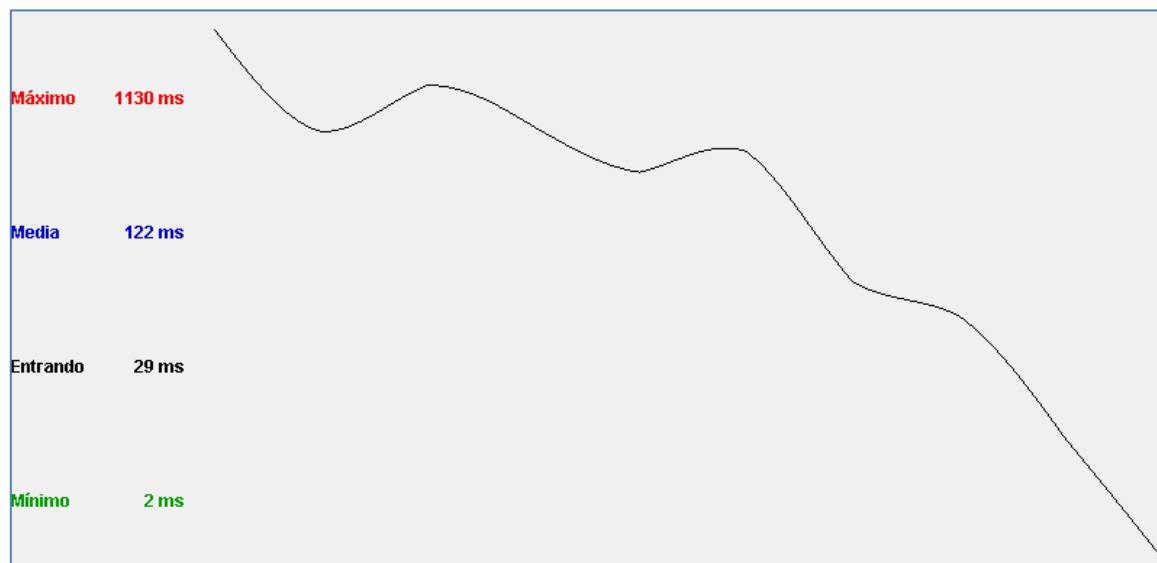


Ilustración 38- Gráfica del tiempo de respuesta de la prueba 5

6.4.6 Prueba 6: 8 grupos de hilos con 3 usuarios por hilo

Informe agregado:

Etiqueta	# Muestras	Media	Mediana	Mín	Máx	Rendimiento	Kb/sec
Home	2400	1377	254	1	59260	2,7	34,5
Ver archivo	1200	1165	102	1	31059	1,4	22,9
Identificarse	600	1057	50	1	31178	0,7	6
Clasificación por tipo	300	799	44	3	30312	0,3	3,9
Clasificación por so	600	801	45	3	30819	0,7	6,7
Descargar archivo	1200	2327	121	2	183300	1,4	221,8
Descargar archivo	300	2082	179	1	91650	0,5	39,5
Búsqueda por palabras	300	1644	325	2	59262	0,4	8,5
Búsqueda avanzada form.	300	4165	274	12	90744	0,4	40,3
Mandar credenciales	600	1400	75	1	59276	0,7	5,5
ver tipo	300	1476	247	0	59256	0,3	10,3
ver so	300	1058	276	63	30487	0,4	10,7
Ver Archivo	900	947	81	1	59258	1	19,5
Búsqueda avanzada	300	832	329	9	31055	0,4	9
Mostrar form. subir archivo	300	811	86	0	30659	0,4	3,6
Subir archivo	300	1185	267	31	30525	0,4	3,1
Ver Windows xp	300	755	187	6	30736	0,3	10,1
Insertar comentario	300	296	40	4	30805	0,3	4,3
Enviar comentario	300	823	72	14	30566	0,3	8,5
Total	11100	1359	165	0	183300	12,4	440,5

Tabla 34- Resultados de la prueba de rendimiento 6

Grafica con el tiempo de respuesta:



Ilustración 39- Gráfica del tiempo de respuesta de la prueba 6

6.5 Resumen y observaciones

Recogemos en una tabla el resumen con los totales de todas las pruebas:

Informe agregado resumen:

Prueba	# Muestras	Media	Mediana	Mín	Máx	Rendimiento	Kb/sec
1 (1 us.)	500	52	9	4	363	2,8	26,9
2 (2 us.)	1000	102	19	4	7452	4,9	125,6
3 (4 us.)	1700	59	19	3	335	9,3	387,3
4 (8 us.)	3700	73	29	2	1230	12,8	464,5
5 (16 us.)	7400	122	94	2	1130	23,5	846,4
6 (24 us.)	11100	1359	165	0	183300	12,4	440,5

Tabla 35- Resumen de los resultados de las pruebas de rendimiento

En términos generales la aplicación se comporta de forma lineal según aumentamos el número de usuarios, es decir, es proporcional el crecimiento de las medidas estudiadas al del número de usuarios, a excepción de la última prueba con 24 usuarios, que nos marca el límite para el hardware que estamos utilizando, y la prueba 2 cuya media y máximo tiempo de respuesta son accidentales, causados por unas pocas muestras muy desviadas, ya que como vemos rendimiento y mediana son correctos.

6.5.1 Tiempo de respuesta

Para el tiempo de respuesta, analizamos la media y la mediana.

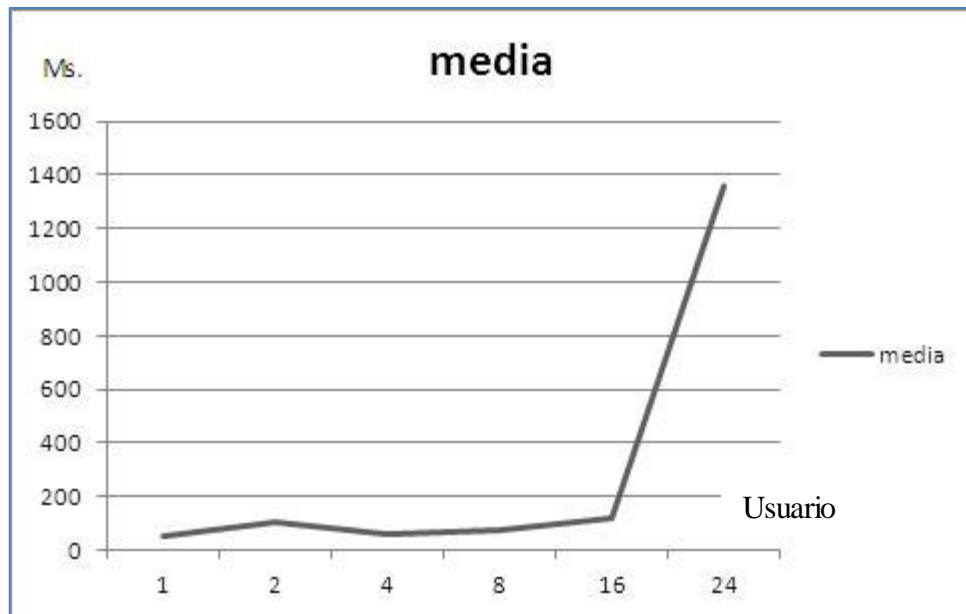


Ilustración 40- Gráfica de la media del tiempo de respuesta

Como vemos se mantiene casi constante con valores menores que 150 ms. excepto para la última prueba (24 usuarios), que como hemos dicho lleva al hardware al límite.

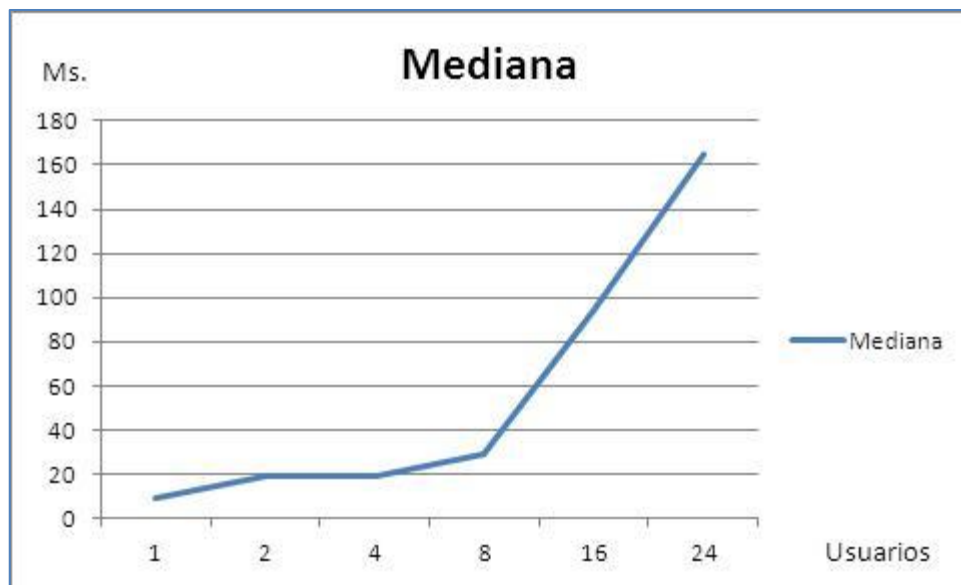


Ilustración 41- Gráfica de la mediana del tiempo de respuesta

La mediana ofrece un crecimiento lineal, con una pendiente más acusada en las dos últimas pruebas, pero el valor no llega a ser excesivamente alto ni siquiera en la última prueba (unos 165 ms.), lo cual nos indica que al menos la mitad de las peticiones se mantienen en tiempos normales incluso en la prueba final, en el límite de nuestra máquina.

6.5.2 Rendimiento



Ilustración 42- Gráfica del rendimiento en muestras por segundo

Como observamos en la gráfica el rendimiento es creciente llegando a su máximo con 16 usuarios y descendiendo con 24 usuarios en el límite de la capacidad del hardware.

7. Planificación y coste del proyecto

En este apartado se expone la planificación para el desarrollo del proyecto, junto al presupuesto estimado para el mismo.

7.1 Planificación

A la hora de planificar lo primero que haremos es dividir el proyecto en tareas. Las tareas contempladas son las siguientes:

- Análisis: comprende las reuniones con el tutor, la determinación de requisitos, análisis de los mismos, estimación, aprendizaje de la tecnología a usar, montaje del entorno.
- Diseño: comprende el modelado por medio del lenguaje UML de los casos de usos determinados en el apartado de análisis.
- Programación: comprende el desarrollo y las pruebas unitarias de los casos de uso.
- Pruebas: comprende las pruebas integrales de los casos de uso y las pruebas de rendimiento.

El proyecto se estima en **150** días de trabajo de 8 horas. **1200** horas de trabajo.

Los 150 días se van a distribuir de la siguiente manera.

Fase	Duración %	Días
Análisis	0,35	52,5
Diseño	0,15	22,5
Programación	0,25	37,5
Pruebas	0,25	37,5

Tabla 36- Distribución del tiempo por fases de desarrollo

Y este es el diagrama de Gantt de la planificación:

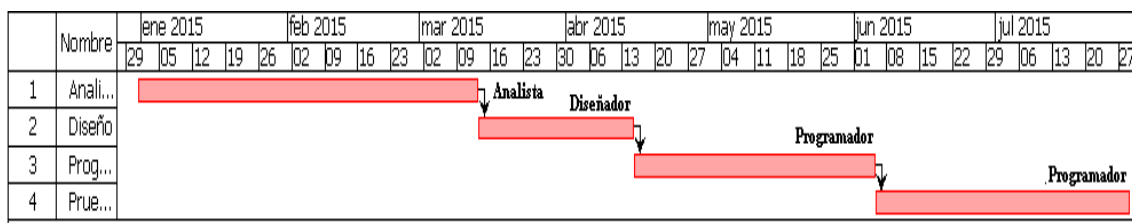


Ilustración 43- Diagrama de Gantt del sistema

7.2 Coste del proyecto

A continuación se detalla cada tipo de coste y el cálculo final con el total del coste del proyecto.

7.2.1 Coste de personal

Para calcular el coste hay que determinar las figuras utilizadas en el proyecto, su correspondencia con las tareas, su coste por hora y las horas que lleva cada tarea.

Las figuras son tres:

- Analista
- Diseñador técnico
- Programador.

Esta es la distribución de tareas.

Tarea	Trabajador
Análisis	Analista
Diseño técnico	Diseñador técnico
Programación	Programador
Pruebas	Programador

Tabla 37- Distribución de tareas por trabajador

Se consideran los siguientes costes por hora por tipo de trabajador. Estos costes vienen a ser frecuentes en el sector en que nos movemos y en el momento concreto en que nos encontramos.

Trabajador	Coste por hora
Analista	55 €
Diseñador técnico	45€
Programador	35€

Tabla 38- Coste en euros por perfil

Calculamos:

Fase	Duración horas	Coste
Análisis	420	23100 €
Diseño	180	8100 €
Programación	300	10500 €
Pruebas	300	4200€
Total	1200	52200€

Tabla 39- Calculo del coste de personal

7.2.2 Coste de los equipos

Necesitamos tres equipos, uno para el analista, otro para el diseñador técnico y otro para el programador, con sus correspondientes monitores. Como precio unitario tomamos 600 euros. Consideramos que en el momento actual por ese precio se puede comprar un equipo con prestaciones más que suficientes para realizar esta tarea. Consideramos un precio unitario para un monitor de 90 euros.

Como no necesitamos los equipos hasta el final de su vida útil, consideramos un plazo de amortización y así podremos ajustar más el gasto. Consideramos un plazo de tres años, que creemos que es razonable. Por último el tiempo de utilización de los equipos es de 7 meses, que es el tiempo que se tarda en desarrollar el proyecto como vimos en la planificación.

A continuación presentamos una tabla resumen del gasto.

Equipos	Cantidad	Coste/Unidad	Total	Amortización	Coste/mes	Coste total
Ordenadores	3	600,00 €	1.800,00 €	3 años	50,00 €	50,00 * 7
Monitores	3	90,00 €	540,00 €	3 años	15,00 €	15,00 * 7
TOTAL	-	-	2340,00 €	-	65,00 €	455,00 €

Tabla 40- Coste de los equipos hardware

7.2.3 Costes indirectos

Los costes indirectos se suelen calcular como un 20 % del coste total.

$$\text{Coste indirecto} = (52200 + 455) \times 0,20 = 10531 \text{ €}$$

7.2.4 Riesgo

Al no ser un proyecto con mucha complejidad técnica se asigna un riesgo bajo de un 5%. Este coste se calcula del total de coste acumulado.

$$\text{Riesgo} = (52200 + 455 + 10531) * 0,05 = 3159,30 \text{ €}$$

7.2.5 Total Coste

Como vemos en la tabla siguiente el coste del proyecto asciende a **80277,81** Euros

Tipo de coste	Coste
Personal	52200,00 €
Equipos	455,00 €
Subtotal	52655,00 €
Indirecto (20%)	10531,00 €
Subtotal	68136,00 €
Riesgo (5%)	3159,30 €
Subtotal	66345,30 €
IVA (21%)	13932,51 €
Total	80277,81 €

Tabla 41- Coste total del proyecto

8. Conclusiones y Trabajos Futuros

8.1 Conclusiones

En las conclusiones hablaremos de los objetivos cumplidos en el producto, los problemas en el proceso, que conocimientos estudiados en la carrera nos han sido útiles, que conocimientos adicionales ha sido necesario obtener.

8.1.1 Producto

Como indicábamos en el apartado “Metodología de desarrollo utilizada”, al adaptar el proceso unificado a nuestro producto proponíamos varias iteraciones y en cada una la realización de una categoría de casos de uso y requisitos en función del tiempo del que se disponía, empezando por los requisitos obligatorios de los casos de uso críticos.

Debido a restricciones de tiempo se han realizado únicamente los requisitos obligatorios de los casos de uso críticos, lo mínimo acordado con el tutor para tener un sistema con una funcionalidad mínima.

Quedarían pendientes para trabajos futuros el resto de iteraciones propuestas, de la numero 3 a la numero 7, como se indica en el apartado de Trabajos Futuros.

Creemos que aunque solo se han realizado los casos de uso críticos, se ha realizado un producto de calidad, poniendo cuidado en obtener un producto fácil de mantener, fácil de ampliar y escalable.

8.1.2 Proceso

Como conclusión en lo que al proceso se refiere, resaltaremos un problema que se ha dado en la planificación y del que hemos aprendido algo muy importante.

Como ya comentábamos en la implementación (ver “Planificación y coste del proyecto”) la parte más difícil en el desarrollo de este sistema ha sido la extracción de datos de versión de los archivos mediante el API de Windows y Javawin.

Con una documentación existente escasa y de no muy buena calidad, este apartado resultó ser más complicado de lo previsto y provocó una desviación en nuestra planificación inicial. En el futuro procuraremos ahondar más en el análisis de aquellas partes poco conocidas y con poca o no muy clara documentación antes de calcular una estimación que debería haber sido más holgada.

8.1.3 Personales

Como conclusiones personales comentaremos que asignaturas de la carrera han sido útiles para la elaboración de este trabajo.

Las asignaturas sobre las que más nos hemos apoyado en la realización de este trabajo son las siguientes:

- Ingeniería del software I: en esta asignatura se estudió el análisis y diseño mediante la metodología tradicional del software. Aunque este proyecto se ha desarrollado mediante la metodología orientada a objetos esta asignatura ha sido útil para la definición de requisitos.
- Ingeniería del software II: en esta asignatura se estudiaron la estimación, el coste, la planificación, el seguimiento, etc. Se han aplicado los métodos vistos en esta asignatura para estimar el tamaño y coste de la aplicación, así como en la planificación.
- Diseño avanzado de software: en ella estudiamos la metodología del Proceso Unificado de Desarrollo, utilizada en el análisis y diseño de este proyecto, y el lenguaje UML, utilizado en los diagramas de diseño.
- Diseño de bases de datos: asignatura que nos facilitó el diseño y desarrollo de la base de datos de la aplicación.
- Asignaturas de programación (Programación I, II y Estructura de datos): Se vio programación estructurada, pero no cabe duda de que fueron de gran ayuda aunque este desarrollo sea orientado a objetos.

Por otra parte mencionar los aspectos específicos aprendidos por cuenta propia que no han sido parte de los estudios de la carrera.

Fue necesario realizar cursos de programación orientada a objetos en lenguaje Java, concretamente se estudiaron cursos de certificación de Sun Microsystems que además de ser útiles para el desarrollo también lo fueron para aplicar la metodología SunTone, marco de este trabajo. Los cursos estudiados fueron los siguientes:

- Java básico
- desarrollo java web y aplicaciones empresariales: **Servlets, JSP y EJB**
- arquitectura y diseño de aplicaciones J2EE.

Por otro lado, parte también fue preciso aprender el framework de base de datos Java **JPA**, con el cual se desarrolló la parte de acceso a datos de la aplicación, en este caso de forma autónoma.

8.2 *Trabajos futuros*

Como indicamos en el apartado de Conclusiones de producto, únicamente se han implementado los requisitos obligatorios de los casos de uso críticos. Quedaría para trabajos futuros la realización del resto de iteraciones propuestas, que eran las siguientes:

Iteración 3: requisitos **obligatorios** de casos de uso **importantes**

Iteración 4: requisitos **obligatorios** de casos de uso **útiles**

Iteración 5: requisitos **opcionales** de casos de uso **críticos**

Iteración 6: requisitos **opcionales** de casos de uso **importantes**

Iteración 7: requisitos **opcionales** de casos de uso **útiles**

8.2.1 Mejoras del prototipo/sistema presentado

Obviamente el sistema se puede mejorar. Por ejemplo se podría haber utilizado un framework de presentación en la capa web, como Struts o Java Server Faces. En su momento fue algo que consideramos que no merecía la pena debido al tamaño del sistema, pero bien es

cierto que un framework de capa web facilitaría posibles ampliaciones, como las propuestas en el apartado Trabajos Futuros.

8.2.2 Otros ámbitos en los que aplicar el trabajo realizado

Este trabajo está centrado en archivos DLL pero se podría aplicar a cualquier tipo de archivo que tenga datos de versión.

Por otro lado tampoco sería difícil incluir librerías de código fuente, cuyo contenido se podría indexar, con lo cual también podríamos hacer búsquedas por el contenido de los comentarios, lo cual sería útil para muchos desarrolladores

En general, con pocas modificaciones este repositorio podría albergar todo tipo de documentos de texto, los cuales podrían ser encontrados por su contenido. El resultado sería algo parecido a una biblioteca digital.

9. Bibliografía

- [1] *DLL y gestión de memoria, guía de programación en Windows*, Mike Klein, ed. Anaya multimedia.
- [2] *DLL-downloads*, <http://www.dll-downloads.com/>
- [3] *Dynamic Linking Library*, <http://es.wikipedia.org/wiki/DLL>
- [4] *DLL Hell*, http://es.wikipedia.org/wiki/DLL_Hell
- [5] *Archivos-DLL*, <http://www.archivos-dll.es/>
- [6] *DLL-downloads*, <http://www.dll-downloads.com/>
- [7] *OO-226- Object Oriented Analysis and Design with UML*, Student Guide, Sun services, Sun Microsystems, modulo 2, pag. 3.
- [8] *UML 2*, Jim Arlow, Ila Neustadt, ed. Anaya Multimedia, pag. 59.
- [9] *UML 2*, Jim Arlow, Ila Neustadt, ed. Anaya Multimedia, pag 83.
- [10] *OO-226- Object Oriented Analysis and Design with UML*, Student Guide, Sun services, Sun Microsystems, modulo 6, pag. 15.
- [11] *Tesaurus*, <http://es.wikipedia.org/wiki/Tesaurus>
- [12] *OO-226- Object Oriented Analysis and Design with UML*, Student Guide, Sun services, Sun Microsystems, modulo 5.
- [13] *Architecting and designing J2EE Applications SL-425*, Student Guide, Sun Services, Sun Microsystems, modulo 13.
- [14] *Architecting and designing J2EE Applications SL-425*, Student Guide, Sun Services, Sun Microsystems, modulo 14.
- [15] *Apache Lucene*, <https://lucene.apache.org/>
- [16] *Jawin, a Java / Win32 Interoperability Project*, <http://jawinproject.sourceforge.net/>
- [17] *Apache Jmeter*, <http://jmeter.apache.org/>

10.Apéndices

10.1 Apéndice 1: Solucionar problemas típicos con DLL

Como decíamos en la introducción es relativamente frecuente que nos aparezca un mensaje de error en Windows sobre un archivo DLL que falta, que no se encuentra o que está dañado, lo cual nos impide ejecutar una aplicación. Para solucionar este problema tendremos que conseguir una copia del archivo y seguir los siguientes pasos:

1. Incluir el archivo en el directorio *system* o *system32* según el sistema operativo, según se indica a continuación:
 - 1.1. XP - C:\Windows\System32.
 - 1.2. 95/98/Me - C:\Windows\System.
 - 1.3. NT / 2000 - C:\WINNT\System32.
2. Reiniciar el equipo.
3. Si todavía hay problemas registramos la DLL. Para ello pulsamos en
“Inicio” -> “Ejecutar” y escribimos:
regsvr32 <archivo>.dll
sustituyendo <archivo> por el nombre del mismo.
4. Si todavía hay problemas copiamos la DLL en el directorio donde está instalado el ejecutable (.exe) de la aplicación [5].

